Final Technical Report
Source Camera Identification and Blind Tamper Detections for Images

AFOSR Contract Number: FA9550-05-1-0130
January 2005 to February 2007

Nasir Memon and Husrev T. Sencar
Polytechnic University
Brooklyn, NY 11201
{memon, hsencar}@poly.edu
(718)-260-3970

1. Cover Sheet: Attached.

2. Objectives: The objective of this project was to develop the science of digital image forensics to a point where authenticity and integrity of digital images can be verified and validated with minimal assumption on the specifics of the generative process. The end goal of the proposed project is to leverage the resources that are available to computer forensics investigators and law enforcement officers, thereby enabling more reliable and accurate decisions on the integrity and authenticity of a digital image prior to admitting it into evidence.

To achieve our goal, we pursued research to integrate and advance current signal and image processing techniques and statistical modeling to develop novel image forensics techniques. Specific objectives of our research included:

Determining the source digital camera of an image. This entailed associating the image with a class of cameras that have common characteristics and matching the image to an individual camera.
Discrimination of synthetic images from real images to identify computer generated images which does not depict a real-life occurrence.
Detection of image tampering to determine whether a given image has undergone any form of modification or processing after it was initially captured.

Several techniques have been proposed to address all these objectives. The involved research involved post-doctoral research associate and graduate students, at the PhD level, in addition to PI. Project deliverables include descriptions of implementation details, image-sets used in experiments and research papers which were delivered to AFRL for experimentation and verification.

REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	April 24th, 2007	Final Technical Report, Jan. 2005 - Feb. 2007

4. TITLE AND SUBTITLE
Source Camera Identification and Blind Tamper Detection Techniques for Images

5. FUNDING NUMBERS
FA9550-05-1-0130

6. AUTHOR(S)
Nasir Memon and Husrev T. Sencar

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Polytechnic University,
Six Metro Tech Center, Brooklyn, NY 11201

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U. S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211
Dr. Robert Herklotz/NM

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AFRL-SR-AR-TR-07-0153

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12 a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)
Digital images can now be easily created, altered, and manipulated with no obvious traces of having been subjected to any of these operations. In this project, we have developed methodologies to verify the authenticity and integrity of digital images in an automatic manner. The results of our project have important implications with regard to ensuring credibility of digital images, especially when it comes to legal photographic evidence. Our proposed techniques can be broadly categorized into three primary areas based on their focus: source camera identification, discrimination of synthetic images, and image forgery detection. In this final technical report we describe our contributions to the field.

14. SUBJECT TERMS
Digital Image Forensics, Tamper/Forgery Detection, Source Camera Identification, Computer Generated Images, Natural Images

15. NUMBER OF PAGES

16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION ON THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

3. Status of Effort

Our objectives towards development of source individual camera and camera-model identification techniques were met by introducing three novel techniques and an improvement over existing approaches. We conducted our experiments on a variety of image data sets which were either generated by our efforts or through crawling public photo sharing sites based on available EXIF headers. We also developed various classification techniques to classify images according to their class characteristics. Our results were published in four conference papers and one journal paper.

Our goal towards developing techniques to discriminate computer generated images from real images has resulted with two novel approaches. In testing our approaches we used large sets of computer generated images that are obtained from specific forums and public sited. Proposed techniques are also incorporated with the existing approaches to improve the performance state-of-the-art techniques.

To achieve our objective of devising tamper detection techniques, we proposed new approaches and tested them rigorously under a variety of settings, compared them with some of the existing approaches, and incorporated them together. However, to determine the true false positive and detection rates of our approach, we need to test them on larger tampered image data sets.

The fact that many proposed image forensics techniques have limitations and none of them by itself offers a definitive solution, have inspired us to continue our work in the field. We are currently studying the problem further both to propose new techniques and combine existing techniques to obtain more reliable decisions. Our ongoing effort will lead to tools that can be utilized by both law enforcement entities and researchers in the field to evaluate and design better techniques

4. Summary of Achievements

We made progress on several fronts in the project. Below we itemize these achievements by topic and summarize the main results obtained. More detailed results are in the papers listed in the publications section.

1. Source Camera-Model Identification

1.1 Image Features: Inspired by the success of universal steganalysis techniques, We proposed a similar approach to identify source camera-model [1]. In essence, a select number of features designed to detect post-processing are incorporated with new features to fingerprint camera-models. The 34 features include color features (e.g., deviations from gray world assumption, inter-band correlations, gamma factor estimates), image quality metrics, and wavelet coefficient statistics. These features are then used to construct multi-class classifiers. The results obtained on moderate to low compressed images taken by 4 different camera-models yielded an identification accuracy of 97%. When experiments are repeated on five cameras where three of them are of the same brand, the accuracy is measured to be 88%. (Tsai et al. [2] later repeated this study using a different set of cameras and reported similar results.) In

[3], we took a similar approach to differentiate between cell-phone camera-models by deploying binary similarity measures as features. In this case, the identification accuracy among nine cell-phone models (of four different brands) is determined as 83%. There are two main concerns regarding this type of approaches. First is that as they provide an overall decision, it is not clear as to what specific feature enables identification which is very important in forensic investigations and in expert witness testimonies. Second concern is the scalability of performance with the increasing number of digital cameras in the presence of hundreds of digital cameras. Hence, in general, this approach is more suitable as a pre-processing technique to cluster images taken by cameras with similar components and processing algorithms.

1.2 CFA and Demosaicing Artifacts: The choice of CFA and the specifics of the demosacing algorithm are some of the most pronounced differences among different digital camera-models. In digital cameras with single imaging sensors, the use of demosacing algorithms is crucial for correct rendering of high spatial frequency image details, and it uniquely impacts the edge and color quality of an image. Essentially, demosaicing is a form of interpolation which in effect introduces a specific type of inter-dependency (correlations) between color values of image pixels. The specific form of these dependencies can be extracted from the images to fingerprint different demosaicing algorithms and to determine the source camera-model of an image. To fingerprint demosaicing algorithms used in different digital camera-models, we deployed expectation/maximization algorithm, assuming a linear model for interpolation within a 5x5 window, and analyzed patterns of periodicity in second order derivates of rows and columns of pixels in moderately smooth and very smooth image parts, respectively [4][5]. The estimated filter coefficients and the periodicity features are used as features in construction of classifiers to detect source camera-model. The accuracy in identifying the source of an image among four and five camera-models is measured as 86% and 78%, respectively, using images captured under automatic settings and at highest compression quality levels.

2. Individual Source Identification

Augmenting Imaging Sensor Imperfections: In [6], Lukas et al. proposed a promising approach to detect the pixel non-uniformity noise, which is the dominant component of the photo-response non-uniformity pattern noise arising due to different sensitivity of pixels to light, to enable source camera matching. To determine the false-positive and true-detection performance of the scheme proposed in [6] under a more realistic setting, we performed experiments on large image data sets and observed that some of the tested cameras yield false-positive rates much higher than the expected values. To better cope with false-positives, we proposed coupling the approach of [7] with camera-model identification methodology. In this case, during the extraction of the pattern the demosaicing characteristics of the source camera-model are also determined as described in [5]. When a decision is to be made in matching an image to a potential source camera, it is also required that the class properties of the camera extracted from the individual image is also in agreement with those of the source camera. It is shown that this approach is very effective in reducing the false-positive rate with a marginal reduction in the true-detection rate.

Sensor Dust Characteristics: We proposed another method for source camera identification based on sensor dust characteristics of single digital single-lens reflex (DSLR) cameras which are becoming increasingly popular because of their interchangeable lenses [8]. Essentially, the sensor dust problem emerges when the lens is removed and the sensor area is opened to the hazards of dust and moisture which are attracted to the imaging sensor due to electrostatic fields, causing a unique dust pattern before the surface of the sensor. Sensor dust problem is persistent and most generally the patterns are not visually very significant. Therefore, traces of dust specks can be used for two purposes: to differentiate images taken by cheaper consumer level cameras and DSLR cameras and to associate an image with a particular DSLR camera. However, it should be noted that the lack of a match between dust patterns does not indicate anything since the dust specks might have been cleaned. Devising an empirical dust model characterized by intensity loss and roundness properties; the authors proposed a technique to detect noise specks on images through match filtering and contour analysis. This information is used in generation of a camera dust reference pattern which is later checked in individual images. In the experiments, ten images obtained from three DSLR cameras are used in generating a reference pattern which is then tested on a mixed set of 80 images (20 taken with the same camera and 60 with other cameras) yielding an average accuracy of 92% in matching the source with no false-positives

3. Identification of Synthetic Images

Motivated by the fact that majority of the real images are captured by digital cameras; we presented an approach that aims at discriminating synthetic images from digital camera images based on the lack of artifacts due to digital camera acquisition process by focusing on the imaging sensor's pattern noise [9]. Although each digital camera has a unique noise pattern, since the underlying sensor technology remains similar, it is very likely that pattern noise introduced by different digital cameras may have common statistical properties. On the other hand, to avoid lack of real-life details, such as textures and lighting, generation of PRCG requires methods that add noise to simulate such phenomena in a physically consistent manner, *e.g.,* ray tracing algorithms. Similarly, it is very likely that the noise introduced by these methods to have certain statistical properties. To test the discriminative ability of the approach, a 600 PRCG images and more than 600 digital camera images have been denoised and the statistics of the resulting noise residues are analyzed. It is shown that the first-order statistics, like skewness and kurtosis, for the two noise components are distinct and the two types of image can be discriminated with an average accuracy of 75%. Later, in [10], we extended this approach to also include demosaicing artifacts [5] and also considered image quality metrics as another set of features to be used for identification. These features are later incorporated with the features of [11-26] some of the other state-of-the-art approaches and tested on 1.1 K PRCG and digital camera images half of which were used for training. Test results show that the classifier designed based on combined features achieves an average accuracy of 93%, which is 5% better than wavelet statistics based features alone [11].

4. Image Forgery Detection

4.1 Variations in Image Features: In this approach designate a set of features that are sensitive to image tampering and determine the ground truth for these features by analysis of original (unaltered) and tampered images. These values are stored as reference values and later tampering in an image is decided based on deviation of the measured features from the ground truth. These approaches most generally rely on classifiers in making decisions. For example, to exploit the similarity between the steganalysis and image manipulation detection, we proposed an approach utilizing image quality metrics to probe different quality aspects of images, which could be impacted during tampering [12]. In [13], image quality metrics are used in cooperation with classifiers to differentiate between original and altered images based on measures obtained between a supposedly modified image and its estimated original (obtained through denoising) in terms of pixel and block level differences, edge distortions, and spectral phase distortions. To ensure that the features respond only to induced distortions due to tampering and not be confused by the variations in the image content, in [12] metrics are also measured with respect to a fixed set of images. Results obtained on 200 images by subjecting them to various image processing operations at a global scale yielded an average accuracy of 80%. When the same classifiers are given 60 skillfully tampered images, the detection accuracy is obtained to be 74%.

Later, we compiled three fundamental sets of features that have been successfully used in universal steganalysis and rigorously tested their sensitivity in detecting various common image processing operations by constructing classifiers to identify images that have undergone such processing [14][15]. The tested features include image quality metrics, wavelet coefficient statistics, binary similarity measures, the joint feature set which combines all the three sets, and the core feature set which is a reduced version of joint feature set. Different types of classifiers built from these features are tested under various image manipulations, like scaling up/down, rotation, contrast enhancement, brightness adjustment, blurring/sharpening and combinations, with varying parameters. Results on 100 locally tampered images, obtained from Internet, show that joint feature set performs best with an identification accuracy of around 90%.

4.2 Inconsistencies in Image Features: Image tampering very often involves local sharpness/blurriness adjustments. Hence, the blurriness characteristics in the tampered parts are expected to differ in non-tampered parts. In [16], we proposed the use of regularity properties of wavelet transform coefficients to estimate sharpness/blurriness of edges to detect variations and to localize tampering. The decay of wavelet transform coefficients across scales has been employed for edge detection and quality estimation purposes previously. The proposed method first employs an edge detection algorithm to determine edge locations which is then followed by a multi-scale wavelet decomposition of the image. Edge locations are located by analyzing the edge image and corresponding maximum amplitude values of wavelet sub-band signals are determined. Then, a linear curve is fitted to the log of these maximum amplitude values and the goodness of the fit is used an indicator of sharpness/blurriness value. The potential of the method in detecting variations in

sharpness/blurriness is demonstrated on both globally blurred images and tampered images with local adjustments.

5. Personnel Supported:

Nasir Memon, H. T. Sencar, Emir Dirik, Sevinc Bayram, and Y. Sutcu.

6. Publications

1 M. Kharrazi, H. T. Sencar, and N. Memon, *Blind Source Camera Identification*, Proc. of IEEE ICIP, 2004.

2 M.-J. Tsai and G.-H. Wu, *Using Image Features to Identify Camera Sources*, Proc. of IEEE ICASSP, 2006.

3 O. Celiktutan, I. Avcibas, B. Sankur and N. Memon, *Source Cell-Phone Identification*, Proc. of ADCOM, 2005.

4 4. S. Bayram, H. T. Sencar and N. Memon, *Source Camera Identification Based on CFA Interpolation*, Proc. of IEEE ICIP, 2005.

5 5. S. Bayram, H. T. Sencar and N. Memon, *Improvements on Source Camera-Model Identification Based on CFA Interpolation*, Proc. of WG 11.9 Int. Conf. on Digital Forensics, 2006.

6 J. Lukas, J. Fridrich and M. Goljan, *Digital Camera Identification from Sensor Pattern Noise*, IEEE Trans. Inf. Forensics and Security, vol. 1, no. 2, pp. 205-214, 2006.

7 Y. Sutcu, S. Bayram, H. T. Sencar and N. Memon, *Improvements on Sensor Noise Based Source Camera Identification*, Proc. of IEEE ICME, 2007.

8 E. Dirik, H. T. Sencar and N. Memon, *Source Camera Identification Based on Sensor Dust Characteristics*, Proc. of IEEE SAFE, 2007.

9 S. Dehnie, H. T. Sencar and N. Memon, *Identification of Computer Generated and Digital Camera Images for Digital Image Forensics*, Proc. of IEEE ICIP 2006.

10 E. Dirik, S. Bayram, H. T. Sencar and N. Memon, *New Features to Identify Computer Generated Images*, Submitted IEEE ICIP, 2007.

11 S. Lyu and H. Farid, Steganalysis Using Higher-Order Image Statistics, IEEE Trans. Image Forensics and Security, vol. 1., no. 1, pp. 111-119, 2006.

12 I. Avcibas, S. Bayram, N. Memon, B. Sankur and M. Ramkumar, *A Classifier Design for Detecting Image Manipulations*, Proc. of IEEE ICIP, 2004.

13 I. Avcibas, B. Sankur and N. Memon, *Steganalysis of Watermarking and Steganography Techniques Using Image Quality Metrics*, IEEE Trans. Image Processing, vol. 12. no. 2, pp. 221-229, 2003.

14 S. Bayram, I. Avcibas, B. Sankur and N. Memon, *Image Manipulation Detection with Binary Similarity Measures*, Proc. of EUSIPCO, 2005.

15 S. Bayram, I. Avcibas, B. Sankur and N. Memon, *Image Manipulation Detection*, Journal of Electronic Imaging, vol. 15, no. 4, 2006.

16 Y. Sutcu, B. Coskun, H. T. Sencar and N. Memon, *Tamper Detection Based on Regularity of Wavelet Transform Coefficients*, Submitted to IEEE ICIP, 2007.

7. **Interactions/Transitions:**
    1. Presented papers at various conferences and workshops.
    2. Presentation to site visit by AFRL in November 2005.
    3. Organized special sessions on Digital Image Forensics in IEEE ICIP 2005 and IEEE SAFE 2007.
    4. Research work described in MIT's Technology Review Magazine, SPIE News Room, Wired Magazine.

1. **New discoveries, inventions, or patent disclosures:** We will explore the possibility of patenting some of the proposed techniques with AFRL.

2. **Honors/Awards:** None.

3. **Markings:** None.

A Classifier Design For Detecting Image Manipulations

Ismail Avcibas, Sevinc Bayram, Nasir Memon, Mahalingam Ramkumar, Bulent Sankur

Department of Electronics Engineering, Uludag University, Bursa, Turkey.
Department of Computer and Information Science, Polytechnic University, Brooklyn, NY, USA.
Department of Electrical and Electronics Engineering, Bogazici University, Istanbul, Turkey.
Department of Computer Science, Mississippi State University, Jackson, MS , USA.

Abstract

In this paper we present a framework for digital image forensics. Based on the assumptions that some processing operations must be done on the image before it is doctored, and an expected measurable distortion after processing an image, we design classifiers that discriminates between original and processed images. We propose a novel way of measuring the distortion between two images, one being the original and the other processed. The measurements are used as features in classifier design. Using these classifiers we test whether a suspicious part of a given image has been processed with a particular method or not. Experimental results show that with a high accuracy we are able to tell if some part of an image has undergone a particular or a combination of processing methods.

1. Introduction

In today's digital age, the creation and manipulation of digital images is made simple by digital processing tools that are easily and widely available. As a consequence, we can no longer take the authenticity of images for granted especially when it comes to legal photographic evidence. *Image forensics*, in this context, is concerned with determining the source and potential authenticity of an image.

Although digital watermarks have been proposed as a tool to provide authenticity to images, it is a fact that the overwhelming majority of images that are captured today do not contain a digital watermark. And this situation is likely to continue for the foreseeable future. Hence in the absence of widespread adoption of digital watermarks, there is a strong need for developing techniques that can help us make statements about the origin, veracity and authenticity of digital images.

In this paper we focus on the problem of reliably discriminating between "doctored" images (images which are altered in order to deceive people) from untampered original ones. The basic idea behind our approach is that a doctored image (or the least parts of it) would have undergone some image processing operations like scaling, rotation, brightness adjustment etc. Hence we first design classifiers that can distinguish between images that have and have not been processed using these basic operations. Then equipped with these classifiers we apply them successively to a suspicious sub-image of a target image and classify the target as doctored if a sub-image classifies differently from the rest of the image.

The rest of this paper is organized as follows: In Section 2 we present a method to compute content independent distortion measure that are used as features in the classifier we design for image forensics. Statistical performance results are given in Section 3, with conclusions drawn in Section 4.

2. Content Independent Features

Our goal is to design a feature based classifier that can discriminate between doctored and original images. The features we use for the classifier should be such that they reflect the distortions an image suffers from manipulation. A classifier based on these statistical features would then differentiate between the two cases of original versus doctored images, even when casual observers cannot perceive them visually. In this section we present a technique for capturing image features that, under some assumptions, are independent of original image content and hence better represent image manipulations.

Now, a doctored image could have been subjected to many operations like scaling, rotation, brightness adjustment, blurring, enhancement etc. or some particular combination thereof. Often doctoring may also involve cutting and pasting of another sub-image, which is skillfully manipulated and rendered along the suture into the original to avoid any suspicion. Since image manipulations can be very subtle, the discriminating features one employs can easily be overwhelmed by variations in the image content.

Keeping the above points in mind it is important to obtain features that remain independent of the image content, so that they would only reflect the presence, if any, of image manipulations. This is due to the fact that in any feature based classification method, there is the risk that the variability in the image content itself may eclipse image alterations present from the detector. Thus, it is desired that whatever features are selected, the detector respond only to *the induced distortions* during doctoring, and not be confused by the statistics of the image content.

In a previous study, we had shown the potential of certain image quality metrics in predicting the presence of steganographic signals within an image [2, 1]. Similar to this approach, we employ mullle image quality metrics as the underlying features of our classifier. The rationale for using mullle quality metrics is to probe different quality aspects of the image, which could be impacted during doctoring manipulations. For example, some measures respond at pixel level, others at the block level, yet others to edge distortions or spectral phase distortion.

Now the main reason image dependence creeps into the classifier is due to the fact that the original image (ground-truth) obviously is not available during the testing stage. Therefore some "ground-truth" or reference signal must be created common to both the training and testing stages. In our previous work on image steganalysis [1], we used a denoised version of the given image as the ground-truth reference. However, creating a reference signal via its own denoised version is obviously a content-dependent scheme.

In the rest of this section, we present an approach to preclude content dependency, by employing a reference image in the feature extraction process. More specifically, let x denote a test image and $x+\varepsilon$ be its processed version, and similarly let y and $y+\eta$ indicate the reference image and is processed version. Furthermore, consider a generic distortion functional $M(a,b)$ between two signals a and b. A simple example of which being the well known mean-square distortion function, $M(a,b) \stackrel{\Delta}{=} E[(a-b)^2]$, with E being the expectation operator. The classifier we design will be based on the statistics of the difference of the distortions, as will be explained in the sequel.

We now make two assumptions for the operation of our classifier. First, we assume the processing operations involved in image doctoring lead to additive distortion, i.e., that is, the altered signals can be represented as $x+\varepsilon$ and $y+\eta$. Second, we assume the additive distortions of the test and reference images are not mutually orthogonal, that is, $E\{\varepsilon^* \eta\} \neq 0$.

We first show that self-referencing, as employed in [1] causes content-dependent distortion. Let f be the specific operation to obtain the reference image; for example in [1] we used a denoising operation. In other words, we had $y = f(x) = denoise(x)$. The outcomes of this operation are given by $x \xrightarrow{f} f(x)$, $x+\varepsilon \xrightarrow{f} f(x+\varepsilon)$, respectively, for original signal and its processed version. To illustrate the point, for the case of the mean-square distortion one obtains:

$$M(x+\varepsilon, f(x+\varepsilon)) - M(x, f(x)) =$$
$$E[f(x+\varepsilon)^2 + 2x\varepsilon + \varepsilon^2 -$$
$$2(x+\varepsilon) f(x+\varepsilon) - 2x f(x) - f(x)^2] \qquad (1)$$

which is content-dependent, because the signal terms x and $f(x)$ survive in the difference of distortion functionals. For content-independence, the above difference should be some function of only the distortion term ε and should not contain x or any of signal derived from it.

Now we take a different route and take as a reference a unique image y. We then measure the distortions between x and $x+\varepsilon$, using y and $y+\eta$ as reference images, $y+\eta$ represents the doctored version of the reference image. The relationship of these signals and the distortion vis-à-vis the reference images y and $y+\eta$ is illustrated in Fig. 1. In this figure, the length of the vector $\vec{xy}$ is simply equal to $M(x,y)$. The distance between the 1s of the vectors $\vec{xy}$ and $\vec{x(y+\eta)}$ is $d = M(x,y) - M(x,y+\eta)$, and similarly $d' = M(x+\varepsilon,y) - M(x+\varepsilon,y+\eta)$ denotes the distance between the 1s of the dashed pair of vectors. For the case of the mean-square distortion it follows that:

$$d = E[(y-x)^2 - (y-x)^2 +$$
$$2(y-x)\eta - \eta^2] = E[2(y-x)\eta - \eta^2]$$

and

$$d' = E[(x+\varepsilon-y)^2 - (x+\varepsilon-y)^2 + 2(x+\varepsilon-y)\eta - \eta^2]$$
$$= E[2(y-x)\eta + 2\eta^*\varepsilon - \eta^2].$$

Now if one considers the difference of d and of d' one can observe that one achieves content-independence, that is:

$$D_1 \stackrel{\Delta}{=} d' - d = 2E[\eta^* \varepsilon] \qquad (2)$$

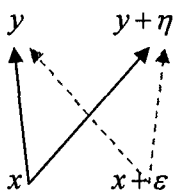


Fig. 1: Configuration of the signal vectors: the original image x, its tampered version $x+\varepsilon$, the reference image y and its tampered version $y+\eta$.

Let's consider another measure, the correlation coefficient, given by $M(a,b) \stackrel{\Delta}{=} E[ab]$. One can easily show that:

$$d = E[xy] - E[x(y+\eta)] = -E[x\eta] \text{ and}$$
$$d' = E[(x+\varepsilon)y] - E[(x+\varepsilon)(y+\eta)] = -E[x\eta] - E[\varepsilon^*\eta)]$$

so that $D_2 \stackrel{\Delta}{=} d' - d = -E[\varepsilon^* n]$. Again the difference of distortions is not a function of image content, x and y, but only of the product of distortions, $\varepsilon^* n$.

We can show that this property holds more generally if the second and higher order partials of the $M(x,y)$ functional are independent of x and y. Consider a generic D:

$$D = M(x,y) - M(x,y+\eta) - M(x+\varepsilon,y) + M(x+\varepsilon,y+\eta)$$

and its variational differential

$$\delta D = -M_x(x,y)\delta x - M_y(x,y)\delta y + M_x(x,y)\delta x + M_y(x,y)\delta y + M_{xy}(x,y)\delta x\delta y \ldots$$

where $M_{x^k y^m}(x,y) = \dfrac{\partial^{k+m} M(x,y)}{\partial x^k \partial y^m}$ [3]. This expression becomes:

$$\delta D = M_{xy}(x,y)\delta x\delta y + high\ \ order\ \ terms \ldots \qquad (3)$$

If the higher order partials of $M(x,y)$ are constant (or zero, as in the cases of the mean-square distortion and correlation coefficient), then the content-independence condition holds.

3. Experimental Results

We selected four measures from the list of image quality measures presented in [1], using Sequential Floating Forward Search (SFFS) algorithm. These three measures, as detailed in the Appendix were the two first-order moments of the angular correlation and two first-order moments of the Czenakowski measure.

We then used a training set of original images and their processed versions, as well as, the original and processed versions of the reference images. We used randomly selected reference images. A linear regression classifier was then designed using the statistics collected with the database of images [3].

The image alterations we experimented with were scaling, rotation, brightness adjustment and contrast enhancement. We trained and tested classifiers for brightness adjustment and contrast enhancement operations separately. In addition, we considered a mixture of alterations, which included scaling, rotation, brightness and contrast enhancements, and designed a classifier for mixed sequential processing. An image database was formed by selecting images from [4] in order to carry out the simulations. The database in [4] contains a rich variety of 2000 images, from which 200 were chosen randomly. Half of the images were used in the training and the remaining in testing.

Table I: The performance of the classifiers

Image Alteration Type	False Positive	False Negative	Accuracy
Brightness Adj.	0/100	23/100	88.5%
Contrast Adj.	6/100	30/100	82%
Mixed Proc.	5/100	12/100	91.5%

The classification accuracies of the detectors designed for specific operations are given in Table I. In these experiments, the entire image was subjected to the same type of operation, as listed in the first column of Table I.

To illustrate how well the selected features capture the impact of the signal processing operations and how well they separate into clusters, we show scatter plots for brightness adjustment, contrast enhancement and mixed sequential processing in Figures 1 a, b and c, respectively. In these figures the axes represent a subset of three features out of the four used. Each figure displays the scattering of the three features obtained from 200 unprocessed (blue), 200 processed (red) images. The axis denoted by d1 and d2 are the standard deviations of angular correlation measure and Czekanowski similarity measures respectively. Third axis d3 is the standard deviation of another correlation measure.

In a second set of more realistic experiments, we addressed the testing of "doctored images". We doctored 16 images by either inserting extra content or replacing the original content. To make them look like natural and avoid any suspicion, the inserted content had to be resized, rotated and brightness adjusted skillfully before pasting it to the image. In some cases we had to blur the block boundaries after pasting. While resizing and rotation were used in every doctored image, we had to do brightness adjustment only in a couple of images. We also obtained 44 doctored images from Internet. We tested 60 doctored images against brightness adjustment, contrast enhancement and mixed sequential processing classifiers. The results of the tests are given in Table II.

Table II: Performance of the classifiers

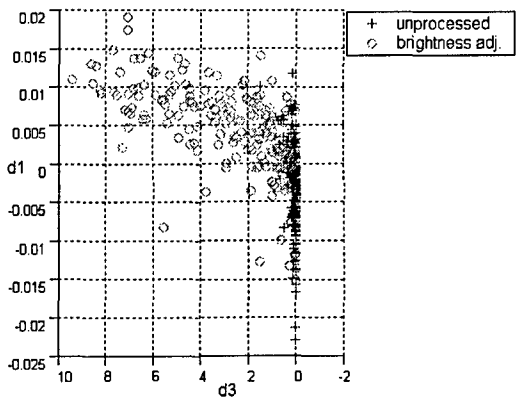
Image Alteration Type	False Positive	False Negative	Accuracy
Brightness Adj.	31/60	3/60	69.2%
Contrast Adj.	25/60	6/60	74.2%
Mixed Proc.	7/60	17/60	80.0%

4. Conclusions

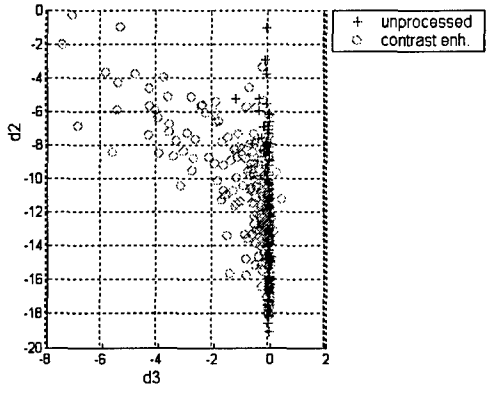
In this paper we proposed a framework for digital image forensics. First, we presented a novel way of content-independent distortion measurement within the framework of image forensics. Second, content-independent distortion measurements were used as features in the design of classifiers. The performance results were encouraging as we were able to discriminate a doctored image from its originals with a reasonable accuracy.

There is significant amount of work that still needs to be done. We need to perform more extensive testing of our classifier. The doctored images we used had a manipulated block sizes that were at least a 100 pixel wide. We need to create test data with smaller manipulations. Also, we need a data set of high quality manipulations as opposed to the ones we generated just for preliminary testing.

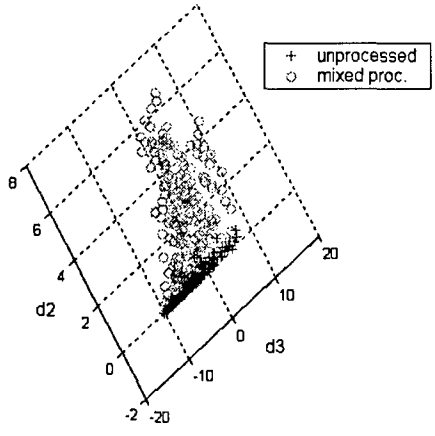
We are also investigating a larger variety of features and the use a more sophisticated classifier as compared to the simple linear classifier we use here.



a)



b)



c)

Figure 1. The scatter diagrams of features in a) brightness adjustment, b) contrast enhencement, c) mixed processing.

5. Appendix

The three different distortion measures used in the paper are shown below. We denote the color components of a three band color image at the pixel position i, j, and in band k as $C_k(i,j)$, where $k = 1,...,3$ and $i, j = 1,..., N$. The boldface symbols $\mathbf{C}(i,j)$ and $\hat{\mathbf{C}}(i,j)$ indicates the color pixel vectors, respectively, of the original and processed image. C itself denotes a color image. The norm and inner product of vectors are defined as

$$\|\mathbf{C}(i,j)\| = \sqrt{C_1(i,j)^2 + C_2(i,j)^2 + C_3(i,j)^2}$$

$$\left\langle \mathbf{C}(i,j), \hat{\mathbf{C}}(i,j) \right\rangle = C_1(i,j)\hat{C}_1(i,j) + C_2(i,j)\hat{C}_2(i,j) + C_3(i,j)\hat{C}_3(i,j)$$

respectively.

First Order Statistics Of Angular Correlation Measure

$$\cos(\Theta_{ij}) = \frac{\left\langle \mathbf{C}(i,j), \hat{\mathbf{C}}(i,j) \right\rangle}{\|\mathbf{C}(i,j)\| \left\|\hat{\mathbf{C}}(i,j)\right\|}, \quad \mu_\theta = \frac{1}{N^2} \sum_{i,j=0}^{N-1} |\cos(\Theta_{ij})|,$$

$$d_1 = \left[\frac{1}{N^2} \sum_{i,j=0}^{N-1} \left(\cos(\Theta_{ij}) - \mu_\theta\right)^2 \right]^{1/2}$$

First Order Statistics of Czekanowski Similarity Measure

$$\chi_{ij} = \frac{2\left\langle \mathbf{C}(i,j), \hat{\mathbf{C}}(i,j) \right\rangle}{\|\mathbf{C}(i,j)\| + \left\|\hat{\mathbf{C}}(i,j)\right\|}, \qquad \mu_\chi = \frac{1}{N^2} \sum_{i,j=0}^{N-1} |\chi_{ij}|,$$

$$d_2 = \left[\frac{1}{N^2} \sum_{i,j=0}^{N-1} \left(\chi_{ij} - \mu_\chi\right)^2 \right]^{1/2}$$

$$\nu_{ij} = \frac{\|\mathbf{C}(i,j)\|}{2\left(\left\|\hat{\mathbf{C}}(i,j)\right\| + \left\langle \mathbf{C}(i,j), \hat{\mathbf{C}}(i,j) \right\rangle \right)},$$

$$\mu_\nu = \frac{1}{N^2} \sum_{i,j=0}^{N-1} |\nu_{ij}|, \quad d_3 = \left[\frac{1}{N^2} \sum_{i,j=0}^{N-1} \left(\nu_{ij} - \mu_\nu\right)^2 \right]^{1/2}$$

6. References

[1] I. Avcibas, N. Memon, B. Sankur, "Steganalysis Using Image Quality Metrics", *IEEE Trans. on Image Processing*, Vol. 12, pp. 221-229, February, 2003.

[2] I. Avcibas, B. Sankur, K. Sayood, "Statistical Evaluation of Image Quality Measures", *Journal of Electronic Imaging*, Vol. 11, pp. 206-223, April, 2002.

[3] A. C. Rencher, *Methods of Multivariate Analysis*, New York, John Wiley (1995).

[4] Image Steganography Database – Dartmouth University. http://www.cs.dartmouth.edu/~farid/.

2004 International Conference on Image Processing (ICIP)

BLIND SOURCE CAMERA IDENTIFICATION

Mehdi Kharrazi [a], *Husrev T. Sencar* [b], *Nasir Memon* [b]

[a] Dept. of Electrical and Computer Eng., Polytechnic University, Brooklyn, NY, USA.
[b] Dept. of Comp. and Inf. Science, Polytechnic University, Brooklyn, NY, USA.

ABSTRACT

An interesting problem in digital forensics is that given a digital image, would it be possible to identify the camera model which was used to obtain the image. In this paper we look at a simplified version of this problem by trying to distinguish between images captured by a limited number of camera models. We propose a number of features which could be used by a classifier to identify the source camera of an image in a blind manner. We also provide experimental results and show reasonable accuracy in distinguishing images from the two and five different camera models using the proposed features.

1. INTRODUCTION

In the analog world, an image (a photograph) has generally been accepted as a "proof of occurrence" of the depicted event. In today's digital age, the creation and manipulation of digital images is made simple by digital processing tools that are easily and widely available. As a consequence, we can no longer take the authenticity of images, analog or digital, for granted. This is especially true when it comes to legal photographic evidence. *Image forensics*, in this context, is concerned with determining some underlying fact about an image. For example image forensics is the body of techniques that attempt to provide authoritative answers to questions such as:

- Is this image an "original" image or was it created by cut and paste operations from different images?
- Was this image captured by a camera manufactured by vendor X or vendor Y?
- Did this image originate from camera X as claimed? At time Y? At location Z?
- Does this image truly represent the original scene or was it digitally tampered to deceive the viewer? For example, was this coffee stain actually a blood stain that was re-colored?
- Was this image manipulated to embed a secret message? That is, is this image a stego-image or a cover-image?

The above questions are just a few examples of issues faced routinely by investigation and law enforcement agencies. However, there is a lack of techniques that could help them in finding authoritative answers. Although digital watermarks have been proposed as a tool to provide authenticity to images, it is a fact that the overwhelming majority of images that are captured today do not contain a digital watermark. And this situation is likely to continue for the foreseeable future. Hence in the absence of widespread adoption of digital watermarks, we believe it is imperative to develop techniques that can help us make statements about the origin, veracity and nature of digital images.

The problems faced in Image Forensics are extremely difficult and perhaps even hard to formulate in a clean and simple manner. In this paper we look at one of the questions above, that is, given an image can we determine the model of the digital camera that was used to capture the image. This is a question that could be often faced during an investigation. Although information about the camera model, type, date and time of the picture are all saved by the camera in the header of the JPEG image, it is not possible to authenticate them. There has been some prior work on identifying the camera used in acquiring a given image [1]. The identification is based on camera characteristics such as defective pixel locations, noise level, image format, and image headers. However such approach is different from the proposed technique in this paper, since it requires the original camera used in image acquisition for evaluation.

The rest of this paper is organized as follows. We start by giving a brief introduction to digital cameras in Section 2. In Section 3, we propose an approach based on feature extraction and classification for the camera source identification problem by identifying a list of candidate features. Experimental results for the two camera case are provided in Section 4. We discuss future work and conclude in Section 5.

This work was supported by AFRL Grant No. F30602-03-C-0091.

0-7803-8554-3/04/$20.00 ©2004 IEEE.

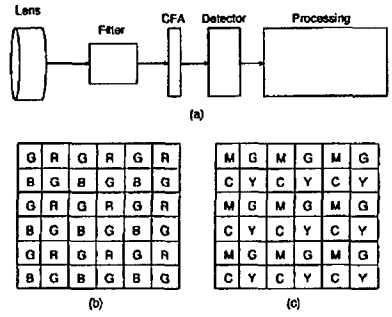


Fig. 1. (a) Major stages of processing in a camera pipeline. (b) CFA pattern using RGB values. (c) CFA pattern using YMCA values

2. DIGITAL CAMERAS

Although much of the details on the camera pipeline are kept as proprietary information of the manufacturer, the general structure and sequence of stages in the camera pipeline seem to be the same in all digital cameras. To set the context for the work presented in later sections, in this section we briefly review the more important stages in a digital camera pipeline. It should be noted that most of the discussion in this chapter is inspired from the introduction to digital cameras by Adams et. al. [2].

The basic structure of a digital camera pipeline can be seen in figure 1(a). After light enters the camera through the lens, a set of filters are employed, the most important being an anti-aliasing filter. The CCD detector is the main component of a digital camera. The detector measures the intensity of light at each pixel location on the detectors surface. In the ideal case, a separate CCD would be used for each of the three color (RGB) channels, but then the manufacturing cost would be quite high. A common approach is to use only a single CCD detector at every pixel, but partition it's surface with different spectral filters. Such filters are called Color Filter Arrays or CFA. Shown in part (b) and (c) of Figure 1 are CFA patterns using RGB and YMCG color space respectively for a 6 6 pixel block. Looking at the RGB values in the CFA pattern it is evident that the missing RGB values need to be interpolated for each pixel. There are a number of different interpolation algorithms which could be used and different manufacturers use different interpolation techniques.

After color decomposition is done by CFA, a detector is used to obtain a digital representation of light intensity in each color band. Next a number of operations are done by the camera, these operations are depicted by the big processor block shown in the figure 1, which include color interpolation as explained before, gamma correction, color processing, white point correction, and last but not least compression. Although the operations and stages explained in this section are standard stages in a digital camera pipeline, the exact processing detail in each stage varies from one manufacturer to the other, and even in different camera models manufactured by the same company. In the next section we will introduce a number of measures which try to capture these differences, and help us in classifying the images originating from a number of cameras.

3. IDENTIFYING MEASURES

One approach to the camera model identification problem is to determine a set of features that designate the characteristics of a specific digital camera, and then use those features to classify obtained images as originating from a specific camera. Although the color image construction process may vary extensively within different makes of digital cameras [2], however, it is our belief that the output image is effected greatly by the following two components:

1. CFA configuration and the demosaicing algorithm
2. The color processing/transformation

As a result of such processing the signal content of the RGB bands will exhibit certain traits and patterns regardless of the original image content. In order to capture the differences in the underlying color characteristics for different cameras we would need to examine the first, second, and possibly higher order statistics of the digital images produced by these cameras. Below we propose a total of 34 features as candidates that would aid in the classification of cameras by make and model:

- *Average pixel value* This measure is based on the *gray world assumption*, which states that the average values in RGB channels of an image should average to gray, assuming that the images has enough color variations. Thus the features are the mean value of the 3 RGB channels (3 features).
- *RGB pairs correlation* This measure attempts to capture the fact that depending on the camera structure, the correlation between different color bands could vary. There are 3 correlation pairs, namely RG, RB (3 features).
- *Neighbor distribution Center of mass* This measure is calculated for each color band separately by first calculating the number of pixel neighbors for each pixel value, where a pixels neighbor are defined as all pixels which have a difference of value of 1 or -1, from

the pixel value in question. The obtained distribution gives us an indication of the sensitivity of the camera pipeline to different intensity levels. We have seen that for a similar image two different cameras have a very similar distribution but one is the shifted version of the other. So we calculated the center of mass of the neighborhood plot to catch that shift as a measure (3 features).

- *RGB pairs energy ratio* is important because it is used in the process of white point correction which is an integral part of a camera pipeline. The calculated features (3 features) are: $E_1 = \frac{|G|^2}{|B|^2}$, $E_2 = \frac{|G|^2}{|R|^2}$,$E_3 = \frac{|B|^2}{|R|^2}$.
- *Wavelet domain statistics* Inspired by Farid's work [3], we decomposed each color band of the image using separable quadratic mirror filters and then calculated the mean for each of the 3 resulting sub-bands (9 features).

In addition to color features, different cameras produce images of different "quality". For example, we commonly notice quality difference between two camera models when images obtained by them are examined visually. For example images obtained by one camera may be sharper but look darker. On the other hand images obtained by another camera may have better lighting and better color quality but are not as sharp as the images obtained by the fist camera. These visual differences that we commonly see motivated us to employ a set of *Image Quality Metrics* (IQM) as features to aid in distinguishing between cameras.

Image Quality Metrics are of utmost importance in providing quantitative data on the quality of a rendered image [4]. IQM's have also been used previously by Memon et al. [5] in the steganalysis problem to distinguish between a set of clean and stego images. We used the same set of IQM's for our studies in this paper. We can categorize the set of IQM's used into three classes based on how the variation between the filtered and original image is measured (13 features):

- the pixel difference based measures (i.e. mean square error, mean absolute error, modified infinity norm);
- the correlation based measures (i.e. normalized cross correlation, Czekonowski correlation);
- the spectral distance based measures (i.e. spectral phase and magnitude errors).

4. EXPERIMENTAL RESULTS

In order to see the effectiveness of the proposed measures in classifying images originating from a digital camera, we conducted a number of experiments. In the first experiment, two different camera models were used, a Sony DSC-P51 and a Nikon E-2100. Both cameras have a resolution of 2 Megapixels. The pictures were taken with maximum resolution, size of 1600 1200, no flash, auto-focus, and the other settings set to the default values. Pictures were taken from the same scene by the two cameras. This is important since for example if one camera was used to take pictures of natural scenery and one camera was used to take pictures of buildings and urban scenery then we might be really detecting the difference in textures of images and not properties due to the camera source.

Fig. 2. The left image was obtained using the Sony DSC-P51, and the right image was obtained by Nikon E-2100.

A picture data set was made by taking 150 pictures with each camera from both inside the university campus buildings as well as other sceneries in New York City; an example is presented in figure 2. Since the Nikon camera had a slightly wider lens, the lens was slightly zoomed at times in order to get the same picture frame as the Sony camera. Only optical zoom was used so that there would be no effects on any of the proposed measures. After collecting the data set, the proposed measures were calculated for each image. A SVM classifier was used in order to see the effectiveness of the proposed features. There are a number of SVM implementations available publicly, and we have used the LibSvm [6] package. A radial basis kernel was used. The following steps were taken in order to design and test the classifier:

1. 2/5 of the 300 images were used in the classifier design phase.
2. The obtained classifier was then used to classify the previously unseen 3/5 of the images.
3. The training and testing steps explained above were repeated 100 times, with a random subset used in each step, in order to see the average classification accuracy.

The average accuracy obtained was 98.73%, and the corresponding confusion matrix could be seen in table 1. In the process of our experiments we also noticed that the quantization table used by each camera was different, further it does also vary from one image to another even with

the same camera. Therefore we re-compressed all images with compression quality set to 75, and then recollected the statistics from the images, designed, and trained the classifier again. The average accuracy was 93.42%, the corresponding confusion matrix could be seen in table 2.

Table 1. The confusion matrix for 2 camera identification case.

		Predicted	
		Nikon	Sony
Actual	Nikon	**99.88**	0.12
	Sony	2.4	**97.6**

Table 2. The confusion matrix for 2 camera identification case after re-compressing the images with JPEG compression quality set to 75.

		Predicted	
		Nikon	Sony
Actual	Nikon	**96.08**	3.91
	Sony	9.25	**90.74**

In the second experiment we wanted to see how the proposed features preform when considering more than two cameras, we obtained 150 images from 3 different models (S100, S110, and S200) of Canon Powershot camera. The images were acquired randomly from the Internet and consist of different sceneries. These 3 models have the same resolution of 2 Megapixels and the images from them have the same size of 1600 1200 (same as the previous 2 cameras studied). However the exact setting used at the time of capture was not known to us. The proposed statistics were collected for the images obtained from the 3 new cameras, and then a multi-class SVM was used to classify data from all of the 5 different camera models, with the same design and testing stages discussed previously. The average accuracy was 88.02%, the corresponding confusion matrix could be seen in table 3. However, we should note that the size and texture diversity of data set being used in the case of 5 cameras, need to be improved for more accurate performance results.

5. CONCLUSION AND FUTURE WORK

In this paper we examined the problem of identifying the source camera of a digital image. Although the problem stated in its full generality is difficult, we looked at a simplified version of the problem where we would like to distinguish between images from a limited number of camera models. As one possible solution we proposed a number of features which could be used in classifying a digital image as originating from a set of digital cameras. A classifier based on these features was then used to see how well the measures could classify the images originating from two cameras used in our experiments. We were also able to achieve acceptable accuracy results after the images were re-compressed.

Table 3. The confusion matrix for 5 camera identification case.

		Predicted				
		Nikon	Sony	Canon (S110)	Canon (S100)	Canon (S200)
Actual	Nikon	**89.67**	0.22	4.77	1.64	3.7
	Sony	3.56	**95.24**	0.31	0.34	0.53
	S110	7.85	0.6	**78.71**	4.78	8.04
	S100	3.14	0.32	3.57	**92.84**	0.11
	S200	5.96	2.27	7.88	0.23	**83.63**

We also showed experimental results with 5 different camera models. Although initial results were encouraging, the true value and performance of the proposed feature set in identifying the camera model would be known when a larger image data set is used. Such a data set needs to be large enough so that the images available from each camera model cover a large range of texture and scenery. Another important research direction is to improve the proposed features which in turn could increase our classification accuracy.

6. REFERENCES

[1] Z. J. Geradts, J. Bijhold, M. Kieft, K. Kurosawa, K. Kuroki, and N. Saitoh, "Methods for identification of images acquired with digital cameras," *Proc. SPIE Vol. 4232, p. 505-512, Enabling Technologies for Law Enforcement and Security*, 2001.

[2] J. Adams, K. Parulski, and K. Spaulding, "Color processing in digital cameras," *Micro, IEEE*, vol. 18, pp. 20–30, Nov.-Dec 1998.

[3] H. Farid and S. Lyu, "Detecting hidden messages using higher-order statistics and support vector machines," *5th International Workshop on Information Hiding.*, 2002.

[4] I. Avcibas, B. Sankur, and K. Sayood, "Statistical evaluation of image quality metrics," *Journal of Electronic Imaging*, April 2002.

[5] I. Avcibas, N. Memon, and B. sankur, "Steganalysis using image quality metrics." *IEEE transactions on Image Processing*, January 2003.

[6] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

SOURCE CAMERA IDENTIFICATION BASED ON CFA INTERPOLATION

Sevinc Bayram [a], *Husrev T. Sencar* [b], *Nasir Memon* [b], *Ismail Avcibas* [a]

[a] Dept. of Electrical and Computer Eng., Uludag University, Bursa, TURKEY
[b] Dept. of Computer and Information Sci., Polytechnic University, Brooklyn, NY, USA

ABSTRACT

In this work, we focus our interest on blind source camera identification problem by extending our results in the direction of [1]. The interpolation in the color surface of an image due to the use of a color filter array (CFA) forms the basis of the paper. We propose to identify the source camera of an image based on traces of the proprietary interpolation algorithm deployed by a digital camera. For this purpose, a set of image characteristics are defined and then used in conjunction with a support vector machine based multi-class classifier to determine the originating digital camera. We also provide initial results on identifying source among two and three digital cameras.

1. INTRODUCTION

The advances in digital technologies have given birth to very sophisticated and low-cost hardware and software tools that are now integral parts of our daily lives. This trend has brought with it new issues and challenges concerning he integrity and authenticity of digital images. The most challenging of these is that digital images can now be easily created, edited and manipulated without leaving any obvious traces of having been modified. This in turn undermines the credibility of digital images presented as news items or as evidence in a court of law since it may no longer be possible to distinguish whether an introduced digital image is the original or not. As a consequence, one can no longer take the authenticity of digital images for granted. Image forensics, in this context, is concerned with determining the source and potential authenticity of a digital image.

Digital watermarking has been introduced as a means for authenticating digital documents that are most likely to undergo various processing [2]. Although this approach enables the extractor to establish the degree of authenticity and integrity of a digital image, it practically requires that the watermark be embedded during the creation of the digital object. This limits watermarking to applications where the digital object generation mechanisms have built-in watermarking capabilities. Therefore, in the absence of widespread adoption of digital watermarks (which is likely to be the case in the (foreseeable future), watermarking cannot be offered as a general solution to the complex problem of authentication. Consequently, in order to determine origin, veracity and nature of digital images, alternative approaches that do not require any prior knowledge of the original digital image need to be considered (blind authentication techniques). *At the present time, however, there is a severe lack of techniques that could achieve these goals*

In this paper, we focus our interest on the source camera identification problem. That is, given an image can we determine the digital camera that was used in capturing the image? It should be noted that all digital cameras encode the camera model, type, date, time, and compression information in the image header; however, it is not possible to authenticate these information. In this regard, the success of blind image authentication techniques rely on the validity of assumption that all images produced by a camera will exhibit certain characteristics, regardless of the captured scene, that are unique to that camera due to its proprietary image formation pipeline. In our prior work [1], we studied the same problem and identified a set of image features by selectively combining the features based on image quality metrics [3] and higher-order statistics of images [4]. This approach essentially requires the design of a classifier that is able to capture the variations in the designated image features, introduced by different digital cameras.

Another promising approach in this area is made by Lukas *et al.* [5]. In their work, sensor's *pattern noise* is characterized via wavelet-based image denoising. The reference noise pattern for each digital camera is obtained by averaging over a number of raw or high quality JPEG images, and the source camera for a given image is determined by correlating the noise pattern with the image itself. Alternatively, in the present work, we exploit the fact that most state-of-the-art digital cameras, due to cost considerations, employ a single mosaic structured *color filter array* (CFA) rather than having different filters for each color component. As a consequence, each pixel in the image has only one color component associated with it, and each digital camera employs a proprietary interpolation algorithm in obtaining the missing color values. Our approach is inspired by the technique proposed by Popescu *et al.* for image tamper detection [6]. The rationale for their

technique is that the process of image tampering very often requires up-sampling operation (which in turn introduces periodic correlations between the image pixels), and they designated statistical measures to detect such phenomena.

The rest of this paper is organized as follows. In section 2, we briefly describe the image formation process in digital cameras. The details for identifying traces of interpolation are provided in Section 3. We present our experimental results in Section 4. and conclude in Section 5.

2. IMAGE FORMATION IN DIGITAL CAMERAS

Although much of the details on the camera pipeline is considered proprietary information to each manufacturer, the general structure and sequence of stages in the camera pipeline remains to be very similar in all digital cameras. The basic structure of a digital camera pipeline is shown in Figure 1-(a) [7]. After light enters the camera through the lens, a set of filters are employed, the most important being an anti-aliasingf ilter. The anti-aliasing filter is needed when the spatial frequency of the scene being captured is larger than the distance between the elements (pixels) of the charge-coupled device (CCD) array.

The CCD array is the main component of a digital camera, and it's the most expensive component. Each light sensing element of CCD array integrates the incident light over the whole spectrum and obtains an electric signal representation of the scenery. Since each CCD element is essentially monochromatic, capturing color images requires separate CCD arrays for each color component. However, due to cost considerations, rather than using multiple arrays, the CCD array is arranged in a pattern by using different spectral filters, typically red, green and blue (RGB). This mask in front of the sensor is called the color filter array (CFA). Since any given CCD element only senses one band of wavelengths, the raw image collected from the array is a mosaic of red, green and blue pixels Figures 1-b and 1-c display a CFA pattern using RGB and YMCG color space respectively for a 6x6 pixel block.

Looking at the RGB values in the CFA pattern, it is evident that each sub-partition of four pixels only provides information on two green, one red, and one blue pixel values. Hence, the missing RGB values need to be interpolated for each pixel (demosaicing). The interpolation is typically carried out by applying a weighting matrix (kernel) to the neighborhood around a missing value. There are a number of different interpolation (demosaicing) algorithms and different manufactories use different interpolation techniques, i.e. kernels with different sizes and shapes. The processing block shown in the Figure 1-a produces the final image and it includes a number of operations which include color processing and compression. Although the operations and stages explained in this section are the standard section of the digital camera pipeline, the exact processing detail in each stage varies from one manufacturer to other, and even in different camera models manufactured by the same company. It should also be noted that many components in the image formation pipeline of various digital cameras, (*e.g.*, lens, optical filters, sensor) are produced by a limited number of manufactures. Therefore, this should be taken into consideration in associating image features with the properties of digital cameras. However, interpolation (demosaicing) algorithm and the design of the CFA pattern remain to be proprietary to each digital camera manufacturer.

In the next section we will describe how the variations in color interpolation can be exploited to classify the images either originating from one camera or the other.

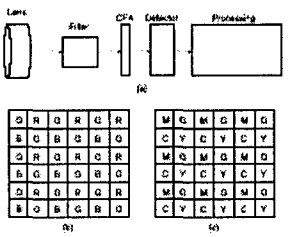


Figure 1. (a) The more important stages of a camera pipeline are shown. (b) CFA pattern using RGB values. (c) CFA pattern using YMCA values.

3. IDENTFYING TRACES OF INTERPOLATION

In [6], Popescu *et al.* employed Expectation/ Maximization (EM) algorithm to detect traces of up-sampling to identify images (or parts of images) that have undergone resizing. The EM algorithm consists of two major steps: an expectation step, followed by a maximization step. The expectation is with respect to the unknown underlying variables, using the current estimate of the parameters, and conditioned upon the observations. The maximization step then provides a new estimate of the parameters. These two steps are iterated until convergence [8]. The EM algorithm generates two outputs. One is a two-dimensional data array, called *probability map*, with each entry indicating the similarity of each image pixel to one of the two groups of samples, namely, the ones correlated to their neighbors and those ones that are not, in a selected kernel. On this *map* the regions identified by the presence of periodic patterns indicate the image parts

that have undergone up-sampling operation. The other output is the estimate of the *weighting (interpolation) coefficients* which designate the amount of contribution from each pixel in the interpolation kernel.

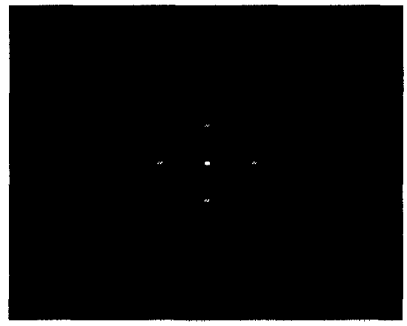
Since in a typical digital camera RGB channels are heavily interpolated, we propose to apply a similar procedure to determine the correlation structure present in each color band and classify images accordingly. Our initial experimental results [1] indicate that both the size of interpolation kernel and the demosaicing algorithm vary from camera to camera. Furthermore, the interpolation operation is highly non-linear, making it strongly dependent on the nature of the depicted scenery. In other words, these algorithms are fine-tuned to prevent visual artifacts, in forms of over-smoothed edges or poor color transitions, in busy parts of the images. On the other hand, in smooth parts of the image, these algorithms exhibit a rather linear characteristic. Therefore, in our analysis we treat smooth and non-smooth parts of images separately.

Since no *a-priori* information is assumed on the size of interpolation kernel (which designates the number of neighboring components used in estimating the value of a missing color component) probability maps are obtained for varying sizes of kernels. When observed in the frequency domain, these probability maps yield to peaks at different frequencies with varying magnitudes indicating the structure of correlation between the spatial samples. In designing our classifier we rely on two sets of features: The set of weighting coefficients obtained from an image, and the peak location and magnitudes in frequency spectrum. In Figure 2, sample magnitude responses of frequency spectrum of the probability maps for three cameras (Sony, Nikon and Canon) are given. The three responses differ in peak locations and magnitudes.

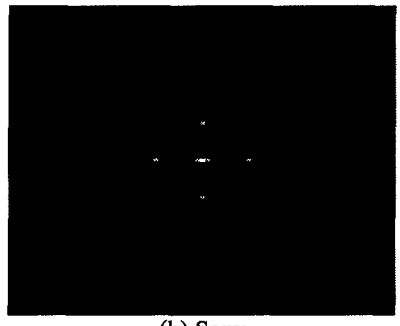
4. EXPERIMENTAL RESULTS

An SVM classifier was used to test the effectiveness of the proposed features. There are a number of SVM implementations available publicly, and we have used the LibSvm package [9]. We have also used the sequential forward floating search (SFSS) algorithm to select the best features from the given set.

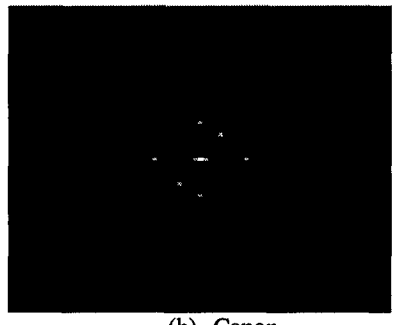
In the first part of our experiments, we have used two camera models: Sony DSC-P51 and Nikon E-2100. The two cameras have both a resolution of 2 mega-pixels. The pictures were taken with maximum resolution, size of 1600x1200 pixels, auto-focus, no focusing, and other settings at default values. In order to detect properties due to the camera source not the texture of images, we used the pictures that were taken from the same scene by two cameras.



(a) Nikon



(b) Sony



(b) Canon

Figure 2. Frequency spectrum of probability maps obtained by three makes of digital cameras.

A picture data set was made by obtaining 140 pictures from each model. One fifth of these images were used for training. Then the designed classifier is used in classifying the previously unseen 4/5 of the images. We used 75x75 pixel parts of theim ages for experiments. First we extracted features assuming a 3x3 interpolation kernel for both Sony and Nikon cameras. The accuracy is measured as 89.3%. The corresponding confusion matrix is given in Table-1.

Table 1. The confusion table for 2 cameras assuming a 3x3 interpolation kernel

		Predicted	
		Nikon	Sony
Actual	Nikon	95.71	4.29
	Sony	17.14	82.86

Then we extract the features considering a neighborhood of 4x4. Correspondingly the accuracy in detection increased to 92.86 and the corresponding confusion matrix is in Table 2. The same experiment is repeated for 5x5 neighborhoods which lead to an accuracy of 95.71%. The corresponding confusion matrix is given in Table 3. As seen from the tables accuracy improves with larger kernel sizes. These results suggest that the actual size of the interpolation kernel used for CFA interpolation is not smaller than the considered sizes which were empirically known to be true [1].

Table 2. The confusion matrix for 2 cameras assuming a 4x4 interpolation kernel

		Predicted	
		Nikon	Sony
Actual	Nikon	91.43	8.57
	Sony	5.71	94.29

Table 3. The confusion tatrix for 2 cameras assuming a 5x5 interpolation kernel

		Predicted	
		Nikon	Sony
Actual	Nikon	94.64	5.36
	Sony	3.57	96.43

In order to see how the proposed features perform when considering three cameras, we also obtained 140 images from Canon Powershot S200. These images were acquired randomly from internet and consist of different sceneries. So we didn't know the exact setting used at the time of capture. Again we used SVM and SFSS to classify three cameras. We extract features from 5x5 neighborhoods. The accuracy was 83.33%, and corresponding confusion matrix is provided in Table 4. Larger neighborhood sizes and new features will be considered in the final version of the paper.

Table 4. The confusion table for 3 cameras assuming a 5x5 interpolation kernel

		Predicted		
		Nikon	Sony	Canon
Actual	Nikon	85.71	10.71	3.57
	Sony	10.71	75	14.28
	Canon	0	10.71	89.28

5. CONCLUSIONS AND FUTURE WORK

In this paper, we propose to identify the source camera of a digital image based on traces of color interpolation in the RGB color channels. For this, we generate a number of measures using EM algorithm. A classifier was then is designed and used to determine how reliably the selected measures could classify the images originating from the two and three cameras.

The proposed approach is another step taken in the direction of devising a set of techniques to solve blind source camera identification problem. This method is, unfortunately, limited to images that are not heavily compressed as the compression artifacts suppress and remove the spatial correlation between the pixels due to CFA interpolation.

6. REFERENCES

[1] M. Kharrazi, H. T. Sencar, N. Memon, "Digital Camera Model Identification," *Proc. of ICIP*, 2004.

[2] Special Issue on Data Hiding, *IEEE Transactions on Signal Processing*, Vol. 41, No. 6, 2003.

[3] I. Avcibas, N. Memon and B. Sankur, "Steganalysis using Image Quality Metrics," *IEEE Transactions on Image Processing*, Jan. 2003.

[4] S. Lyu and H. Farid, "Detecting Hidden Messages Using Higher-Order Statistics and Support Vector Machines," *Proc. of Information Hiding Workshop*, 2002

[5] J. Lukas, J. Fridrich, and M. Goljan, "Determining Digital Image Origin Using Sensor Imperfections," *Proc. of IS&T SPIE*, vol 5680, 2005

[6] A. Popescu and H. Farid, "Exposing Digital Forgeries by Detecting Traces of Re-sampling," *IEEE Transactions on Signal Processing*, 2004.

[7] J. Adams, K. Parulski and K. Sapulding, "Color Processing in Digital Cameras," *IEEE Micro*, Vol. 18, No.6, 1998.

[8] Todd Moon, 'The Expectation Maximization Algorithm', *IEEE Signal Processing Magazine, November 1996.*

[9] C. Chang and C. Lin, "LIBSVM: A library for support vector machines," 2001, Software available at http://www.csie.ntu.edu.tw/»cjlin/libsvm

IMAGE MANIPULATION DETECTION WITH BINARY SIMILARITY MEASURES

Sevinç Bayram[a], İsmail Avcıbaş[a], Bülent Sankur[b], Nasir Memon[c]

a Department of Electronics Engineering, Uludağ University, Bursa, Turkey.
b Department of Electrical and Electronics Engineering, Boğaziçi University, İstanbul, Turkey.
c Department of Computer and Information Science, Polytechnic University, Brooklyn, NY, USA.

sevincbayram@yahoo.com, avcibas@uludag.edu.tr, sankur@boun.edu.tr, memon@poly.edu

ABSTRACT

Since extremely powerful technologies are now available to generate and process digital images, there is a concomitant need for developing techniques to distinguish the original images from the altered ones, the genuine ones from the doctored ones. In this paper we focus on this problem and propose a method based on the neighbor bit planes of the image. The basic idea is that, the correlation between the bit planes as well the binary texture characteristics within the bit planes will differ between an original and a doctored image. This change in the intrinsic characteristics of the image can be monitored via the quantal-spatial moments of the bit planes. These so-called Binary Similarity Measures are used as features in classifier design. It has been shown that the linear classifiers based on BSM features can detect with satisfactory reliability most of the image doctoring executed via Photoshop tool.

Keywords: Digital image forensics, image processing, binary similarity measures, classification.

1. INTRODUCTION

The advances in digital technologies have given birth to very sophisticated and low-cost tools that are now integral parts of information processing. This trend brought with it new challenges concerning the integrity and authenticity of digital documents, in particular images. The most challenging of these is that digital images can now be easily created, edited and manipulated without leaving any obvious traces of having been modified. As a consequence, one can no longer take the authenticity of images for granted, especially when it comes to legal photographic evidence. Image forensics, in this context, is concerned with determining the source and potential authenticity of a digital image.

Digital watermarks can serve in a scheme to authenticate images. However, presently the overwhelming majority of images that circulate in the media and Internet do not contain a digital watermark. Hence in the absence of widespread adoption of digital watermarks or concurrently with it, we believe it is necessary to develop image forensic techniques. We define image forensics as the art of reconstituting the set of processing operations, called overall doctoring, that the image has been subjected to. In turn these techniques will enable us to make statements about the origin, veracity and nature of digital images.

In a prior work [6], we studied the same problem of reliably discriminating between "doctored" images (images which are altered in order to deceive people) from untampered original ones. The detection scheme was based on training a classifier based on certain image quality features, called also "generalized moments". Scaling, rotation, brightness adjustment, blurring, enhancement etc. or some particular combinations of them are typical examples of doctoring. A frequent image manipulation involves the pasting of another image, skillfully manipulated so to avoid any suspicion. Since the image manipulations can be very subtle to eschew detection, the discriminating features can be easily overwhelmed by the variation in the image content. It is, thus, very desirable to obtain features that remain independent of the image content, so that they would only reflect the presence, if any, of image manipulations.

2. BINARY SIMILARITY MEASURES

We assume that altering an image changes the correlation between and within bit planes. Therefore the quantal-spatial correlation between the bit planes of the original image will differ from that of the bit planes of the doctored images. Consequently certain statistical features extracted from the bit planes of images can be instrumental in revealing the presence of image manipulations. Since each bit plane is also a binary image, we start by considering similarity measures between two binary images. These measures, called Binary Similarity Measures (BSM) were previously employed in the context of image steganalysis.[1, 3]. In this paper we measure the correlation between bit planes numbered 3-4, 4-5, 5-6, 6-7 and 7-8 for the red channel and bit planes 5-5 of the red and blue channels.

Classical measures are based on the bit-by-bit matching between the corresponding pixel positions of the two images. Typically, such measures are obtained from the scores based on a contingency table (or matrix of agreement) summed over all the pixels in an image. In this study, we have found that it is more relevant to make comparison based on *binary texture statistics*. Let $\mathbf{x}_i = \{x_{i-k}|,\ k = 1,\ldots,K\}$ and $\mathbf{y}_i = \{y_{i-k}|,\ k = 1,\ldots,K\}$ be the

sequences of bits representing the K-neighborhood pixels, where the index i runs over all the MxN image pixels. For K=4 we obtain the four stencil neighbors and for K=8 we obtain the 8 neighbors. Let

$$c_{r,s} = \begin{cases} 1 & if\ x_r = 0 \quad and \quad x_s = 0 \\ 2 & if\ x_r = 0 \quad and \quad x_s = 1 \\ 3 & if\ x_r = 1 \quad and \quad x_s = 0 \\ 4 & if\ x_r = 1 \quad and \quad x_s = 1 \end{cases} \tag{1}$$

Then we can define the agreement variable for the pixel x_i as: $\alpha_i^j = \sum_{k=1}^{K} d(c_{i,j-k}, j), \quad j = 1...4, \quad K = 4$, where

$$\delta(m,n) = \begin{cases} 1 & , \quad m = n \\ 0 & , \quad m \neq n \end{cases}. \tag{2}$$

The accumulated agreements can be defined as:

$$a = \frac{1}{MN}\sum_i \alpha_i^1, \quad b = \frac{1}{MN}\sum_i \alpha_i^2,$$
$$c = \frac{1}{MN}\sum_i \alpha_i^3, \quad d = \frac{1}{MN}\sum_i \alpha_i^4. \tag{3}$$

These four variables {a,b,c,d} can be interpreted as the one-step co-occurrence values of the binary images. Obviously these co-occurrences are defined for a specific bit plane b, though the bit plane parameter was not shown for the sake simplicity. Normalizing the histograms of the agreement scores for the b[th] bit-plane (where now $a_i^j = a_i^j(b)$) one obtains for the j'th co-occurrence:

$$p_j^\beta = \sum_i \alpha_i^j \Big/ \sum_i \sum_j \alpha_i^j; \quad \beta = 3...8 \tag{4}$$

In addition to these we calculate the Ojala [4] texture measures as follows. For each binary image on the b[th] bit-plane we obtain a 256-bin histogram based on the weighted K=8 neighborhood as in Fig. 1. For each 8-neighborhood pattern, the histogram bin numbered $n = \sum_{k=0}^{7} x_{i-k} 2^i$ is augmented by one.

1	2	4
128	x_i	8
64	32	16

(a)

0	1	0
1	x_i	0
0	1	1

(b)

Fig. 1 (a) The weighting of the neighbors in the computation of Ojala score. (b) An example: Ojala score S=2+16+32+128=178

Let the two normalized histograms be denoted as S_n^b, $n = 0...255$ and $b = 3..7$. The resulting Ojala measure is the mutual entropy between the two distributions belonging to adjacent planes b and $b+1$:

$$m_b = -\sum_{n=1}^{N} S_n^b \log S_n^{b+1}. \tag{5}$$

Table I. Binary Similarity Measures

Similarity Measure	*Description*
Sokal & Sneath Similarity Measure 1	$m_1 = \frac{a}{a+b} + \frac{a}{a+c} + \frac{d}{b+d} + \frac{d}{c+d}$
Sokal & Sneath Similarity Measure 2	$m_2 = \frac{ad}{\sqrt{(a+b)(a+c)(b+d)(c+d)}}$
Sokal & Sneath Similarity Measure 3	$m_3 = \frac{2(a+d)}{2(a+d)+b+c}$
Sokal & Sneath Similarity Measure 4	$m_4 = \frac{a}{a+2(b+c)}$
Sokal & Sneath Similarity Measure 5	$m_5 = \frac{a+d}{b+c}$
Kulczynski Similarity Measure 1	$m_6 = \frac{a}{b+c}$
Ochiai Similarity Measure	$m_7 = \sqrt{\left(\frac{a}{a+b}\right)\left(\frac{a}{a+c}\right)}$
Binary Lance and Williams Nonmetric Dissimilarity Measure	$m_8 = \frac{b+c}{2a+b+c}$
Pattern Difference	$m_9 = \frac{bc}{(a+b+c+d)^2}$
Binary Minimum Histogram Difference	$dm_{10} = \sum_{n=1}^{4} \boldsymbol{min}(p_n^\beta, p_n^{\beta+1})$
Binary Absolute Histogram Difference	$dm_{11} = \sum_{n=1}^{4} \lvert p_n^\beta - p_n^{\beta+1} \rvert$
Binary Mutual Entropy	$dm_{12} = -\sum_{n=1}^{4} p_n^\beta \boldsymbol{log}\, p_n^{\beta+1}$
Binary Kullback Leibler Distance	$dm_{13} = -\sum_{n=1}^{4} p_n^\beta \boldsymbol{log} \frac{p_n^\beta}{p_n^{\beta+1}}$
Ojala Minimum Histogram Difference	$dm_{14} = \sum_{n=1}^{N} \boldsymbol{min}(S_n^\beta, S_n^{\beta+1})$
Ojala Absolute Histogram Difference	$dm_{15} = \sum_{n=1}^{N} \lvert S_n^\beta - S_n^{\beta+1} \rvert$
Ojala Mutual Entropy	$dm_{16} = -\sum_{n=0}^{15} S_n^\beta \boldsymbol{log}\, S_n^{\beta+1}$
Ojala Kullback Leibler Distance	$dm_{17} = -\sum_{n=1}^{N} S_n^\beta \boldsymbol{log} \frac{S_n^\beta}{S_n^{\beta+1}}$

We have used three types of binary similarity measures between bit planes as in Table 1.

First group:. The measures m_1 to m_9 are obtained for neighbor bits separately by applying the parameters moments $\{a,b,c,d\}$ in (3) to the binary string similarity measures, such as Sokal & Sneath.

Second group: The differences $dm_i = m_i^\beta - m_i^{\beta+1}$ $i = 10,\ldots,13$ are used as the final measures.

Third group: Measures dm_{14}-dm_{17} are the neighborhood-weighting mask proposed by Ojala [4].

3. EXPERIMENTAL RESULTS

We computed binary similarity measures as features and used Sequential Floating Forward Search (SFFS) algorithm to select the best features [5] and we have used Linear Regression Classifier for classification [7]. In our experiments we have built a database of 200 images. These images were taken with Canon Powershot S200 camera. Notice that the images that were taken from the same camera in order to detect alterations, but not the properties due to the camera characteristics.

The image alterations we experimented with were scaling-up, rotation, brightness adjustment, blurring and sharpening, all implemented via Adobe Photoshop [8]. Half of the images were used for training and the remaining in testing. In [2], Farid *et al.* employed a higher order statistical model to discriminate natural images from unnatural ones. We have adopted their method, so that we did the same tests once with their features and then with our features. In the Table's below the results according to features in [2] are denoted as "Farid". First, we scaled-up all the images with the scales of %50, %25, %10, %5, %2, %1 and got 6 databases of 200 images. We trained a classifier on each database and tested if an image is original or scaled-up. The results are in Table II.

Table II. The performance for image scaling-up attack.

Scaling-up	Method	False Positive	False Negative	Accuracy (%)
%50	BSM	2/100	0/100	99
	Farid	4/100	11/100	92.5
	Farid	5/100	11/100	92
%10	BSM	18/100	3/100	89.5
	Farid	4/100	17/100	89.5
%5	BSM	25/100	4/100	85.5
	Farid	4/100	14/100	91
	Farid	8/100	21/100	85.5
%1	BSM	32/100	8/100	80
	Farid	17/100	12/100	85.5

We rotated the images 45°, 30°, 15°, 5°, 1°. Corresponding results are in Table III.

Table III. The performance for rotation attack.

Rotation	Method	False Positive	False Negative	Accuracy (%)
%50	BSM	2/100	0/100	99
	Farid	4/100	11/100	92.5
%25	BSM	7/100	0/100	96.5
	Farid	5/100	11/100	92
%10	BSM	18/100	3/100	89.5
	Farid	4/100	17/100	89.5
%5	BSM	25/100	4/100	85.5
	Farid	4/100	14/100	91
%2	BSM	27/100	7/100	83
	Farid	8/100	21/100	85.5

We adjusted the brightness of the images with the scales of 40, 25, 15, 5. Corresponding results are in Table IV.

Table IV. The performance for brightness adjustment attack.

Brightness Adjustment	Method	False Positive	False Negative	Accuracy (%)
40	BSM	17/100	27/100	78
	Farid	60/100	28/100	58
25	BSM	13/100	32/100	77.5
	Farid	61/100	26/100	56.5
15	BSM	19/100	28/100	76.5
	Farid	67/100	27/100	53.5
5	BSM	18/100	45/100	68.5
	Farid	59/100	39/100	51

We use Gaussian blur to blur the images with the scales of 1, 0.5, 0.3, 0.1. Corresponding results are represented in Table-V.

Table V. The performance for blurring attack.

Blurring	Method	False Positive	False Negative	Accuracy (%)
1.0	BSM	1/100	0/100	99.5
	Farid	0/100	7/100	96.5
0.5	BSM	2/100	0/100	99
	Farid	81/100	1/100	59
0.3	BSM	46/100	22/100	66
	Farid	49/100	38/100	56.5
0.1	BSM	24/100	62/100	57
	Farid	69/100	31/100	50

We sharpen the images and train a classifier to distinguish the sharpened ones from the original ones. In Table VI, we show the results of the sharpening classifier.

Table VI. The performance for sharpening attack.

	Method	False Positive	False Negative	Accuracy (%)
Sharpening	BSM	4/100	9/100	93.5
	Farid	36/100	19/100	72.5

As shown in the tables we trained more than one classifier for each image alteration type at different settings of attack strength. However, it is not practical to devise a separate classifier for each setting; hence we trained one classifier per alteration type to operate in a range of attack strengths. For example we generate an image pool with 50 images from %25, %10, %5, and % 2 scaled-up. We used half of the images for training and remained for testing. The results for generic classifier for various image alteration types are given in Table VII.

To test an image on only one classifier we made an image pool by adding the same quantity of images that are scaled up with the scales of %50, %25, %10, %5, scaled down %50, %25, %10, %5, rotated 45°, 30°, 15°, 5°, contrast enhanced with the scales of 25,15,5, brightness adjusted with the scales of 15, 25, blurred with the scales of 0.3, 0.5 and sharpened. Again half of the images were used for training and the remaining for testing. We call this classifier as generic-generic classifier. Corresponding results for this classifier is shown in Table VIII.

Table VII. The performance of generic classifiers.

Image Alteration Type	Method	False Positive	False Negative	Accuracy (%)
Scaling Up	BSM	12/100	3/100	92.5
	Farid	6/100	17/100	88.5
Scaling Down	BSM	29/100	13/100	79
	Farid	17/100	18/100	82.5
Rotation	BSM	13/100	45/100	71
	Farid	16/100	14/100	85
Contrast Enhancement	BSM	1/100	48/100	75.5
	Farid	79/100	13/100	54
Brightness Adjustment	BSM	3/100	46/100	75.5
	Farid	76/100	17/100	53.5
Blurring	BSM	6/100	18/100	88
	Farid	80/100	4/100	58

Table VIII. The performance of generic-generic classifiers.

Method	False Positive	False Negative	Accuracy (%)
BSM	21/100	28/100	75.5
Farid	15/100	31/100	77

To make our results more realistic, we addressed the testing of "doctored images". We doctored 20 images by either inserting extra content or replacing the original content. To make them look like natural and avoid any suspicion, the inserted content was resized, rotated or brightness adjusted etc, before pasting it to the image. We take 2 untampered and one tampered block from every image, so we had 40 untampered and 20 tampered blocks. We tested these blocks on generic classifiers. We accept it as tampered if any of the generic classifiers declare it as tampered. In Table IX the results for the image blocks on generic classifiers are shown.

Table IX. The perf. of generic classifiers for image blocks.

Method	False Positive	False Negative	Accuracy (%)
BSM	9/40	2/20	81.67
Farid	40/40	0/20	33.3

And we tested the same blocks on generic - generic classifiers. The corresponding results are in Table X.

Table X. The perf. of generic classifiers for image blocks.

Method	False Positive	False Negative	Accuracy (%)
BSM	8/40	4/20	80
Farid	9/40	8/20	71.67

We capture 100 images from Internet that can easily be tampered. We tested these images on generic and generic - generic classifiers. The results are shown in Table XI and Table XII.

Table XI. The performance of generic classifiers for image blocks that are captured from Internet.

Method	False Negative	Accuracy
BSM	9/100	91
Farid	0/100	100

Table XII. The performance of generic-generic classifiers for image blocks that are captured from internet.

Method	False Negative	Accuracy (%)
BSM	48/100	52
Farid	47/100	53

4. CONCLUSIONS

In this paper we proposed a method for digital image forensics, based on Binary Similarity Measures between bit planes used as features. Then we designed several classifiers to test the tampered or un-tampered status of the images. The performance results in detecting and differentiating a host of attacks were encouraging as we were able to discriminate a doctored image from its original with a reasonable accuracy. We have assessed our methods vis-à-vis the closest competitor image forensic detector in [2]. We outperform Farid's detector especially in contrast enhancement and brightness adjustment attacks. On the other hand, while we have better performance at stronger levels of manipulations, Farid outperforms us at weaker levels. In this respect, the two schemes seem to be complementary; hence fusion of forensic detectors at feature level or decision level must be envisioned.

5. REFERENCES

[1] Avcıbaş, İ., N. Memon, B. Sankur. 2002. Image Stegalysis with Binary Similarity Measures, Proceedings of International Conference on Image Processing, Volume 3, 645-648, 2002.

[2] Farid, H., S. Lyu. Higher-Order Wavelet Statistics and their Application to Digital Forensics, IEEE Workshop on Statistical Analysis in Computer Vision (in conjunction with CVPR), Madison, Wisconsin, 2003.

[3] İ. Avcıbaş, I., M. Kharrazi, N. Memon, B. Sankur, Image Steganalysis with Binary Similarity Measures, Applied Signal Processing (under review)

[4] Ojala, T., M. Pietikainen, D. Harwood. A Comparative Study of Texture Measures with Classification Based on Feature Distributions, Pattern Recognition, vol.29, pp, 51-59.

[5] Pudil, P., F. J. Ferri, J. Novovicov and J. Kittler. Floating search methods for feature selection with nonmonotonic criterion functions. In Proceedings of the 12th ICPR, volume 2, pages 279-283, 1994.

[6] Avcıbaş, İ., S. Bayram, N. Memon, M. Ramkumar, B. Sankur, A Classifier Design for Detecting Image Manipulations, Proceedings of International Conference on Image Processing, 2004, Singapore.

[7] A. C. Rencher, *Methods of Multivariate Analysis*, New York, John Wiley (1995).

[8] www.adobe.com

Source Cell-phone Identification

Oya Çeliktutan[a], İsmail Avcıbaş[a], Bülent Sankur[b], Nasir Memon[c]
[a]*Uludag University, Electronics Engineering Department, Bursa, Turkey*
[b]*Bogazici University, Electrical-Electronics Engineering Department, Istanbul, Turkey*
[c]*Dept. of Comp. and Inf. Science, Polytechnic University, Brooklyn, NY, USA*
oyaxceliktutan@hotmail.com, avcibas@uludag.edu.tr, bulent.sankur@boun.edu.tr, memon@poly.edu

Abstract

The techniques to validate the authenticity of digital images are rather limited. In this paper, we focus on blind source cell-phone identification problem. The main idea is that proprietary interpolation algorithm (involved due to the structure of color filter array [CFA]) leaves footprints in the form of correlations across adjacent bit planes of the image. For this purpose, we define a set of binary similarity measures and image quality measures in conjunction with a KNN classifier to identify the originating cell-phone. We provide results on identifying source among three cell-phones.

1. Introduction

Image forensics is a new emerging field concerned with determining the source and potential authenticity of a digital objects and possibly reconstructing the history of manipulations effected. In this sense image forensics tries to meet the new challenge of safeguarding the authenticity of digital image and to enable their continued usefulness as trustworthy documents and legal evidence. Digital images can obviously be easily created, edited and manipulated with increasingly more sophisticated tools, which do not leave much of any perceptible trace.

Digital watermarking falls short to meet all desiderata of this particular problem [1]. On the one hand, watermarking requires that imaging devices be equipped with built-in watermarking capabilities; on the other hand, watermarks may not be able to classify all types of attack. Forensic tools, however, can be envisaged to identify the nature of the manipulation. Finally, forensic tools can be used concomitantly with watermarking in decision fusion schemes.

In this paper, we focus on the identification of source cell-phones. In other words, the problem is to determine the make and the brand of the camera with which the given image was captured. The camera brand/made identification is based on the telltale effects due the proprietary image formation pipeline. In fact, the main difference between cameras originates from the color filter array that is used to interpolate between color pixels. In prior works [2], [3], source camera identification problem was studied using feature sets based on image quality metrics [4] and higher-order statistics [5].

All camera identification techniques exploit the fact that state-of-the-art cell-phone cameras, due to cost considerations, employ a single mosaic structured *color filter array* (CFA) rather than having different filters for each color component [6]. This process is illustrated in Fig. 1. As a consequence each pixel in the image has only one color component associated with it, and each digital camera employs a proprietary interpolation algorithm in obtaining the missing color values. This very proprietary interpolation algorithm leaves footprint like correlations between contiguous bit planes of an image.

In this work we use binary symmetry features, which directly address correlation properties within and between planes. We consider also mixtures of other categories of features, such Image Quality Measures (IQM) [4]. The rest of this paper is organized as follows. In section 2, we briefly describe the similarity measures used in the classifier design, which were selected from a set of measures described in [4], [7]. The details of the technique and experimental results are provided in Section 3. We discuss future work and present our conclusions in Section 4

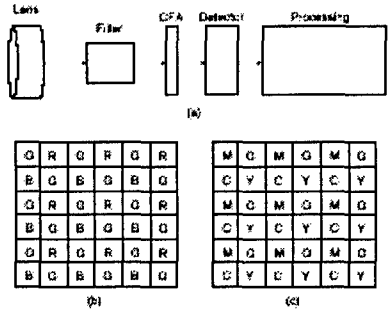


Fig. 1. (a) The more important stages of a camera pipeline are shown [6]. (b) CFA pattern using RGB values. (c) CFA pattern using YMCA values.

2. Similarity Measures

Since each bit plane is also a binary image, we start by considering similarity measures between two binary images, that is, between quantal bit planes of images. The binary similarity measures were extensively studied in [7]. We discuss here two of them for illustrative purposes.

Let's consider the 5-point stencil function and apply it in the bit plane *b*:

$$a_c^n(k,b)=\begin{cases} 1 & if \quad x_c=0 \quad and \quad x_n=0 \\ 2 & if \quad x_c=0 \quad and \quad x_n=1 \\ 3 & if \quad x_c=1 \quad and \quad x_n=0 \\ 4 & if \quad x_c=1 \quad and \quad x_n=1 \end{cases}$$

where the four arguments are defined as follows: The subscript c defines some central pixel and the superscript n denotes one of the possible four neighbor pixels. We sum $a_c^n(k,b)$ over its four neighbors (i.e. n runs over East, West, South and North neighbors) as well as over all the pixels (i.e., c runs over the MxN pixels). After the summations the sub- and superscripts can be omitted. The first argument *k indicates* one of the four agreement scores {1,2,3,4} and the second argument indicates the bit plane in which this computation is being done. Obviously $\{a(k,b),k=1,..,4\}$ variables, that is, the agreement scores the central pixel – neighbor pixel transition types in a particular bit plane. Normalizing the agreement scores we obtain the score pdf's:

$p_k^b=\alpha(k,b)/\sum_k \alpha(k,b)$. Based on these normalized *four-bin* histograms, we define binary Kullback Leibler distance as:

$$m_1=-\sum_{n=1}^{4} p_n^7 \log \frac{p_n^7}{p_n^8}.$$

The second measure m_2 is somewhat different in that we use the neighborhood-weighting mask proposed by Ojala [8]. The histogram of the Ojala moments for different cameras is plotted in Fig. 4. For each binary image we obtain a 512-bin histogram based on the weighted neighborhood, where the score is given by: $S=\sum_{i=0}^{7} x_i 2^i$ by weighting the eight directional neighbors as shown in Fig. 2. Defining S_n^7 the count of the n^{th} histogram bin in the 7^{th} bit plane and S_n^8 the corresponding one in the 8^{th} plane, after normalizing these 512-*bin* histograms, we can define absolute histogram difference as:

$$m_2=\sum_{n=0}^{511}\left|S_n^7-S_n^8\right|.$$

1	2	4
128	256	8
64	32	16

Fig. 2: The weighting pattern of the neighbors in the computation of Ojala score. For example, the score becomes S=2+ 4+8=14 in the example where E, N, NE bits are 1 and all other bits are 0.

The image quality measures were extensively studied in [4]. We discuss here one of them for illustrative purposes. The Czenakowski distance gives a metric useful to compare vectors with strictly non-negative components, as in the case of color images:

$$m_3=\frac{1}{MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\left(1-\frac{2\sum_{k=1}^{3}\min\left(C_k(i,j),\hat{C}_k(i,j)\right)}{\sum_{k=1}^{3}\left(C_k(i,j)+\hat{C}_k(i,j)\right)}\right),$$

where $C_k(i,j)$ is $(i,j)^{th}$ pixel of the k^{th} band of a color image and $\hat{C}_k$ is the denoised version of the corresponding k^{th} band color image. Denoising is employed on the image to obtain a reference image to calculate the metric.

In Fig. 3 we give the scatter plot of three cell-phone cameras for three features, namely, m1, m2, m3 measures. As can be seen the used features cluster well enough for a successful classification.

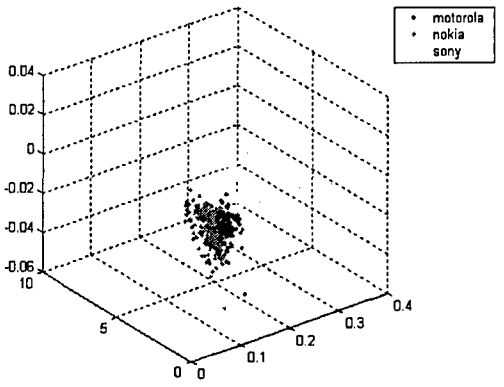


Fig. 3. The scatter plot of three cell-phone cameras for three similarity measures.

Overall we considered 108 BSM features and 10 IQM features. The BSM features consisted of the 7-8, 6-7, 5-6, 4-5, 3-4 bit planes of the red channel and of the 5th bit plane of the remaining blue and green channels. These features were then selected using the Sequential Forward Feature Selection (SFFS) algorithm.

3. Experimental Results

We have considered nine makes and/or brands of cell phone cameras, as detailed in Table 1:

Table 1. Types of cameras tested and their display characteristics.

Acronym	Make/Brand	Colors	Resolution (pixel)
M1	MotorolaV3	260K	176 x 220
M2	MotorolaV500	65K	176 x 220
N1	Nokia5140	65,536	128 x 128
N2	Nokia6230	65,536	208 x 208
N3	Nokia6600	65,536	176 x 208
N4	Nokia7270	65,536	128 x 160
S1	SonyK700	65,536	176 x 220
S2	SonyK750	262,144	176 x 220
L1	LG5600	65K	128 x 160

We collected 200 images from each one of them with maximum resolution, size of 640X480 pixels, at day light and auto-focus mode. Half of the 1800 images are used for training and the designed classifier is tested with the other unseen half set of images. The images were typical shots varying from nature scenes to close-ups of people. We experimented with the KNN classifier (K=5) as well SVM algorithm of RBF variety (γ=2.0,ε=0.001,C=8.0,cachesize=40). Sample images of outdoors scenes in the image database are shown in Fig. 4.

In a first exploratory experiment, we grouped cameras in three-tuples and ran SFFS algorithm for each combination for the best selection of features. Sample confusion tables from these three-cameras groups of are given below (best, middle, worst case tables given):

Table 2a. Confusion matrix for the SonyK700, MotorolaV3, Nokia6230 group. SFFS resulted in 5 features. Overall performance = 98.7%.

	SonyK700	MotorolaV3	Nokia6230
SonyK700	100	0	0
MotorolaV3	0	100	0
Nokia6230	0	4	96

Table 2b. Confusion matrix for the SonyK750, MotorolaV3, Nokia6600 group. SFFS resulted in 3 features. Overall performance = 90.0%.

	SonyK750	MotorolaV3	Nokia6600
SonyK750	92	8	0
MotorolaV3	8	87	5
Nokia6600	1	8	91

Table 2c. Confusion matrix for the SonyK750, MotorolaV3, Nokia7270 group. SFFS resulted in 7 features. Overall performance = 81.3%.

	SonyK750	MotorolaV3	Nokia7270
SonyK750	71	6	23
MotorolaV3	1	97	2
Nokia7270	18	6	76

The average performance of all 16 different three-tuple experiments was 93.4%.

In a more challenging experiment we tried to classify the pool of nine camera types. Again the SFFS

algorithm was run for the ensemble of camera classes with the SVM classifier. The results are given in Table 3. The overall accuracy fell to 62.3, which is still considerably better than the random guess results of 11.1%.

Table 3: Recognition performance of camera brands in pairwise comparisons.

	S1	S2	M1	M2	N1	N2	N3	N4	L1
S1	92	4	0	0	0	0	0	3	0
S1	5	63	1	0	4	0	0	32	3
M1	0	3	60	11	3	1	3	3	5
M2	0	0	22	67	4	9	5	0	5
N1	0	6	2	3	57	3	6	7	18
N2	0	1	2	6	3	68	22	0	0
N3	0	1	7	10	1	18	62	1	1
N4	3	16	2	0	7	0	0	36	12
L1	0	6	4	3	21	1	2	18	56

4. Conclusions and Future Work

In this work, we proposed to identify the originating cell-phone of a digital image based on the combination of binary similarity measures, computed across contiguous bit planes of an image, and of image quality measures. The apposite features were selected based on the SFFS algorithm and an SVM classifier was trained for classification.

The performance of small groups of camera makes/brands is very satisfactory. The classification performance in groups of two is close to 100% and in groups of three it scores around 93%. For the larger group of 9 cameras, a classifier attains 62.3% correct classification.

This study can be advanced along several avenues. A larger set of features, including the so-called Higher Order Statistical measures as in [5,9]. We have just considered the red channel in this work. The perturbation of the correlation structure across color channels as well as within the blue and green channels remains to be investigated. Fusion techniques, especially the sum or product rule variety, has been shown to be very effective in improving classifier performance [10]. Another aspect of our study is to measure the drop in the performance when the technique is applied on medium- and low-quality (better compressed) images. Another direction will be to utilize binary similarities on all color channels and use more sophisticated classifiers like support vector machines.

Acknowledgement: This work has been supported in part by TÜBİTAK under the research grant number 104E056.

5. References

[1] Special Issue on Data Hiding, *IEEE Transactions on Signal Processing*, Vol. 41, No. 6, 2003.

[2] M. Kharrazi, H. T. Sencar, N. Memon, "Digital Camera Model Identification," *Proc. of ICIP*, 2004.

[3] S. Bayram, H. T. Sencar, N. Memon, I. Avcibas, "Source Camera Identification Based on CFA Interpolation", ICIP 2005.

[4] I. Avcibas, N. Memon and B. Sankur, "Steganalysis using Image Quality Metrics," *IEEE Transactions on Image Processing*, Jan. 2003.

[5] S. Lyu and H. Farid, "Detecting Hidden Messages Using Higher-Order Statistics and Support Vector Machines," *Proc. of Information Hiding Workshop*, 2002.

[6] J. Adams, K. Parulski and K. Sapulding, "Color processing in digital cameras," *IEEE Micro*, Vol. 18, No.6, Jun. 1998.

[7] I. Avcibas, M. Kharrazi, N. Memon, B. Sankur, "Image Steganalysis with Binary Similarity Measures", *Journal of Applied Signal Processing*, in press, 2005.

[8] T. Ojala, M. Pietikainen, D. Harwood, A Comparative Study of Texture Measures with Classification Based on Feature distributions, Pattern Recognition, vol. 29, pp. 51-59.

[9] Popescu and H. Farid ,"Exposing Digital Forgeries by Detecting Traces Of Re-sampling", IEEE Transactions on Signal Processing ,2004.

[10] J. Kittler, F.M. Alkoot, Sum Versus Vote Fusion in Multiple Classifier Systems, IEEE Trans. Patttern Recognition and Machine Intelligence, 25, 110–115, 2003.

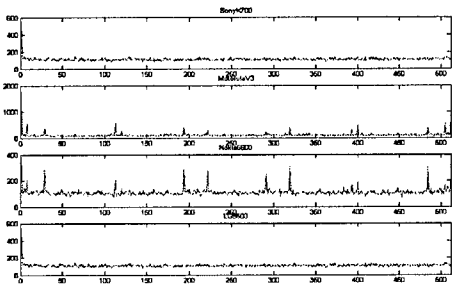


Fig. 4: Plot of Ojala histograms for different cameras.

Chapter 1

IMPROVEMENTS ON SOURCE CAMERA-MODEL IDENTIFICATION BASED ON CFA INTERPOLATION

Sevinc Bayram, Husrev T. Sencar and Nasir Memon

Abstract The idea of using traces of interpolation algorithms, deployed by a digital camera, as an identifier in the source camera-model identification problem has been initially studied in 2. In this work, we improve our previous approach by incorporating methods to better detect the interpolation artifacts in smooth image parts. To identify the source camera-model of a digital image, new features that can detect traces of low-order interpolation are introduced and used in conjunction with a support vector machine based multi-class classifier. Performance results due to newly added features are obtained considering source identification among two and three digital cameras. Also, these results are combined with those of 2 to further improve our methodology.

Keywords: Network forensics, wide area networks

1. Introduction

The advances in digital technologies have given birth to very sophisticated and low-cost hardware and software tools that enabled easy creation, distribution and modification of digital images. This trend has brought with it new challenges concerning he integrity and authenticity of digital images. As a consequence, one can no longer take the authenticity of digital images for granted. Image forensics, in this context, is concerned with determining the source and potential authenticity of a digital image.

Although, digital watermarking technologies 3 have been introduced as a measure to address this problem, its realization requires that the watermark be embedded during the creation of the digital image. Essentially, this necessitates digital cameras to have built-in watermarking

capabilities. However, this approach has not been adopted by digital camera manufacturers. Consequently, to determine origin, veracity and nature of digital images, alternative approaches need to be considered. The setting of this problem is further complicated by the requirements that the methods should require as little as possible prior knowledge on the digital camera and the actual conditions under which the image has been captured (blind image authentication). At the present time, there is a severe lack of techniques that could achieve these goals.

The underlying assumption for the success of blind image authentication techniques is that all images produced by a digital camera will exhibit certain characteristics regardless of the captured scene, which are unique to that camera, due to its proprietary image formation pipeline. It should be noted that all digital cameras encode the camera model, type, date, time, and compression information in the EXIF image header. However, since this information can be easily modified or removed, it cannot be used for authentication.

In this paper, we concentrate on source camera-model identification problem by identifying the traces of proprietary interpolation algorithm deployed by digital cameras. For this, we improve our results in 2 by incorporating new methodologies to capture CFA interpolation artifacts due to low-order interpolation.

The rest of this paper is organized as follows. In the following section, existing approaches to image source identification problem are discussed. In section 3, we briefly describe the image formation process in digital cameras. In Section 4, the results of 2 are reviewed, and the details of the improved approach are provided. We present our experimental results in Section 5 and conclude in Section 6.

2. Current Solutions

In our prior work 1, we studied the source camera-model identification problem by identifying and selectively combining a set of image features based on image quality metrics 4 and higher-order statistics of images 5. This approach essentially requires the design of a classifier that is able to capture the variations in the designated image features, due to different digital cameras.

Another promising approach in this area is made by Lukas et al. 6. In their work, an imaging sensor's pattern noise is characterized via wavelet-based image denoising. The reference noise pattern for a particular digital camera is obtained by averaging obtained noise residual over a number of high quality JPEG images captured by that camera. Then, a given image is matched to a camera by correlating the noise pattern

of the particular camera (which is claimed to be used for capturing the image in question) with the individual noise pattern extracted from the image itself.

In 2, we exploit the fact that most state-of-the-art digital cameras, due to cost considerations, employ a single mosaic structured color filter array (CFA) rather than having different filters for each color component. As a consequence, each pixel in the image has only one color component associated with it, and each digital camera employs a proprietary interpolation algorithm in obtaining the missing color values for each pixel. Our approach in 2 was inspired by the technique proposed by Popescu et al. intended for image tamper detection 7. The rationale for their technique is that the process of image tampering very often requires up-sampling operation which in turn introduces periodic correlations between the image pixels. To detect such phenomena they designated statistical measures. In a similar manner, we have applied variants of such measures to characterize the specifics of the deployed interpolation algorithm.

In the present work, we further improve our approach in 2 by designating new features. Due to perceptual image quality considerations, designers have to tailor the interpolation algorithm to deal with different qualities in an image, i.e., edges, texture features, etc. This essentially requires introducing strong non-linearities to the interpolation algorithm. However, in relatively smooth image parts, most well known interpolation algorithms (e.g., bilinear and bicubic methods) will ensure satisfactory quality, and very expensive algorithms are not needed. Our premise in this work is that most proprietary algorithms in smooth image parts will deploy simpler forms of interpolation, and therefore, they can be captured more effectively (as opposed to busy image parts where interpolation requires more careful processing). For this purpose, we utilize the results of 8 where the periodicity pattern in the second order derivative of interpolated signal is analyzed.

3. Image Formation in Digital Cameras

The structure and sequence of processing stages of image formation pipeline in a digital camera remains to be very similar in all digital cameras (despite the proprietary nature of the underlying technology). In a digital camera, the light entering the camera through the lens is first filtered (the most important being an anti-aliasing filter) and focused onto an array of charge-coupled device (CCD) elements, i.e., pixels. The CCD array is the main and most expensive component of a digital camera. Each light sensing element of CCD array integrates the incident light

over the whole spectrum and obtains an electric signal representation of the scenery. Since each CCD element is essentially monochromatic, capturing color images requires separate CCD arrays for each color component. However, due to cost considerations, in most digital cameras, only a single CCD array is used by arranging them in a pattern where each element has a different spectral filter, typically one of red, green or blue (RGB). This mask in front of the sensor is called the color filter array (CFA). Hence, each CCD element only senses one band of wavelengths, and the raw image collected from the array is a mosaic of red, green and blue pixels.

As each sub-partition of pixels only provide information about a number of green, red, and blue pixel values, the missing RGB values for each pixel need to be obtained through interpolation (demosaicing). The interpolation is typically carried out by applying a weighting matrix (kernel) to the neighboring pixels around a missing value. Most generally, each manufacturer uses a proprietary demosaicing algorithm i.e., kernels with different sizes, shapes and different interpolation functions. This is followed by a processing block which typically involves a number of operations like color processing and compression to produce a faithful representation of the scenery being imaged.

Although the image formation pipeline remains same for almost all cameras, the exact processing detail at all stages vary from one manufacturer to other, and even in different camera models manufactured by the same manufactures. It should also be noted that many components in the image formation pipeline of various digital cameras, (e.g., lens, optical filters, CCD array) are produced by a limited number of manufactures. Therefore, due to this overlap, different cameras may exhibit similar qualities, and this should be taken into consideration in associating image features with the properties of digital cameras. However, interpolation (demosaicing) algorithm and the design of the CFA pattern remain to be proprietary to each digital camera manufacturer. In the next section we will describe how the variations in color interpolation can be exploited to classify the images either originating from one camera or the other.

4. Identifying Traces of Interpolation

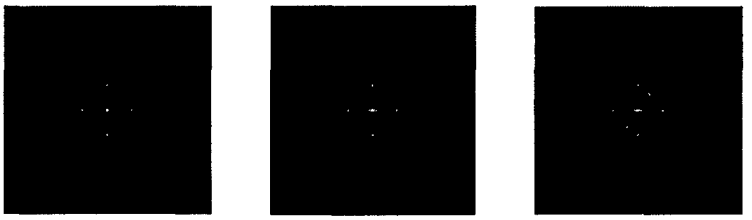
In 7, Popescu et al. presented a methodology to detect traces of upsampling to identify images (or parts of images) that have undergone resizing by analyzing the correlation of each pixel value to its neighbors. Since in a typical digital camera RGB channels are heavily interpolated, we proposed to apply a similar procedure to determine the correlation

structure present in each color band and classify images accordingly 2. Our initial experimental results indicate that both the size of interpolation kernel and the demosaicing algorithm vary from camera to camera 1. Furthermore, the interpolation operation is highly non-linear, making it strongly dependent on the nature of the depicted scenery. In other words, these algorithms are fine-tuned to prevent visual artifacts, in forms of over-smoothed edges or poor color transitions, in busy parts of the images. On the other hand, in smooth parts of the image, these algorithms exhibit a rather linear characteristic. Therefore, in our analysis we treat smooth and non-smooth parts of images separately.

4.1 Non-smooth Image Parts

We employ Expectation/ Maximization (EM) algorithm to detect traces of interpolation 7. The EM algorithm consists of two major steps: an expectation step, followed by a maximization step. The expectation is with respect to the unknown underlying variables, using the current estimate of the parameters, and conditioned upon the observations. The maximization step then provides a new estimate of the parameters. These two steps are iterated until convergence 10. The EM algorithm generates two outputs. One is a two-dimensional data array, called probability map, in which each entry indicate the similarity of each image pixel to one of the two groups of samples, namely, the ones correlated to their neighbors and those ones that are not, in a selected kernel. On this map the regions identified by the presence of periodic patterns indicate the image parts that have undergone up-sampling operation. The other output is the estimate of the weighting (interpolation) coefficients which designate the amount of contribution from each pixel in the interpolation kernel.

Since no a priori information is assumed on the size of interpolation kernel (which designates the number of neighboring components used in estimating the value of a missing color component) probability maps are obtained for varying sizes of kernels. When observed in the frequency domain, these probability maps yield to peaks at different frequencies with varying magnitudes indicating the structure of correlation between the spatial samples. In designing our classifier we rely on two sets of features: The set of weighting coefficients obtained from an image, and the peak location and magnitudes in frequency spectrum. In Figure 1, sample magnitude responses of frequency spectrum of the probability maps for three cameras (Sony, Nikon and Canon) are given. The three responses differ in peak locations and magnitudes.



(a) (b) (b)

Figure 1. Frequency spectrum of probability maps obtained for **(a)** Nikon E-2100, **(b)** Sony DSC-P51 and **(c)** Canon Powershot S200 digital cameras.

4.2 Smooth Image Parts

In 8, Gallagher showed that low-order interpolation introduces periodicity in the variance of the second order derivative of an interpolated signal which can be subsequently used to determine the interpolation rate and algorithm of the signal. The proposed interpolation detection algorithm first obtains the second order derivative of each row and averages it over all rows. When observed in the frequency domain the locations of the peaks reveal the interpolation rate and the magnitude of the peaks determine the interpolation method.

We employed a similar methodology to characterize the interpolation rate and the method employed by a digital camera. It should be noted that most digital cameras encode and compress images in JPEG format. Due to 8x8 block coding, the DC coefficients may also introduce peaks in the second-order derivative implying the presence of some form of interpolation operation at a rate of 8. Therefore, in detecting the interpolation algorithm, the peaks due to JPEG compression have to be ignored. Figure 2 displays the magnitude frequency response for the three models of digital cameras. The variation in magnitude and indicates that there are differences in the deployed interpolation algorithm. Therefore, the features extracted from each camera include the location of the (peaks except for the ones due to JPEG compression), their magnitudes, and the energy of each frequency component with respect to other frequency components at all color bands.

5. Experimental Results

An SVM classifier was used to test the effectiveness of the proposed features. There are a number of publicly available SVM implementations. Our work is based on the LibSvm package 11. We have also used

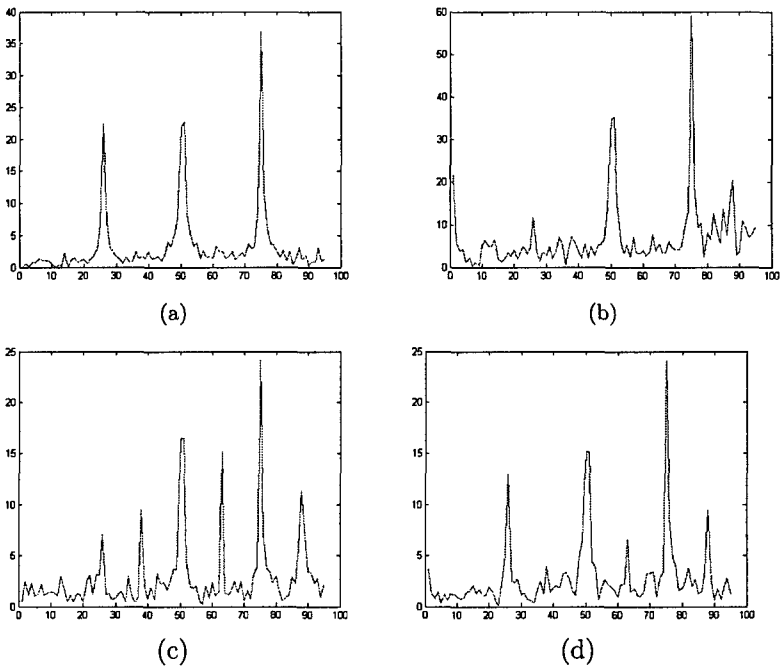


Figure 2. Frequency spectrum of averaged second order derivatives corresponding to **(a)** JPEG compression and the three models of digital cameras, **(b)** Canon Powershot S200, **(c)** Sony DSC-51, **(d)** Nikon E-2100, with JPEG output images.

the sequential forward floating search (SFSS) algorithm to select the best features from a given set of features.

In the first part of our experiments, we have used two camera models: Sony DSC-P51 and Nikon E-2100. The two cameras have both a resolution of 2 mega-pixels. The pictures are of size 1600x1200 pixels and are obtained with maximum resolution, auto-focus, and other settings at default values. In order to reduce the dependency on the scenery being viewed, we used pictures that were taken from the same scene by two cameras. A picture data set was made by obtaining 140 pictures from each model. One third of these images were used for training. Then the designed classifier is used in classifying the previously unseen 2/3 of the images. We used 75x75 pixel parts of the images for experiments. Based on the variance of each block the image is partitioned into smooth and non-smooth parts by an exhaustive search.

First we extracted features assuming a 3x3 interpolation kernel for both Sony and Nikon digital cameras. The accuracy is measured as

Table 1. The confusion table for 2 cameras assuming a 3x3 interpolation kernel

		Predicted	
		Nikon	Sony
Actual	Nikon	95.7%	4.3%
	Sony	17.1%	82.9%

Table 2. The confusion table for 2 cameras assuming a 4x4 interpolation kernel

		Predicted	
		Nikon	Sony
Actual	Nikon	91.4%	8.6%
	Sony	5.7%	94.3%

Table 3. The confusion table for 2 cameras assuming a 5x5 interpolation kernel

		Predicted	
		Nikon	Sony
Actual	Nikon	94.6%	5.4%
	Sony	3.6%	96.4%

89.3%. Then, we extracted the features considering a neighboring 4x4 pixels. Correspondingly the accuracy in detection increased to 92.86%. The same experiment is repeated for 5x5 neighborhoods which lead to an accuracy of 95.71%. The corresponding confusion matrices are given in Tables 1, 2, and 3, respectively. As seen from the tables accuracy improves with larger kernel sizes. These results suggest that the actual size of the interpolation kernel used for CFA interpolation is not smaller than the considered sizes which were empirically known to be true 1. Similar performance results are also obtained from smooth image parts using the features based on periodicity in the second order derivatives. Table 4 displays the accuracy for the two camera case. It is seen that the latter set of features do not prove as reliable as the former set of features.

In order to see how the proposed features perform for the case of three-cameras, we also obtained a set of images acquired by a Canon Powershot S200. In this case, the images were downloaded from internet and consist of different sceneries. In a similar manner, we extracted the

Table 4. The confusion matrix for 2 cameras based on periodicity in the second-order derivative

		Predicted	
		Nikon	Sony
Actual	Nikon	86.9%	13.1%
	Sony	23.3%	76.7%

Table 5. The confusion table for 3 cameras assuming a 5x5 interpolation kernel

		Predicted		
		Nikon	Sony	Canon
Actual	Nikon	85.7%	10.7%	3.6%
	Sony	10.7%	75%	14.3%
	Canon	0%	10.7%	89.3%

Table 6. The confusion table for 3 cameras based on periodicity in the second-order derivative

		Predicted		
		Nikon	Sony	Canon
Actual	Nikon	76.8%	8.9%	14.3%
	Sony	12.5%	76.8%	10.7%
	Canon	19.6%	10.7%	69.6%

features described in Sections 3.1-2 and used SVM and SFSS to classify three cameras. When features are extracted from 5x5 neighborhoods, the accuracy is measured as 83.33%, and corresponding confusion matrix is provided in Table 5. When attempted to discriminate cameras on the basis of features obtained from smooth image parts, the accuracy dropped to 74.3% as shown in Table 6.

Finally, we have combined the two sets of features and repeated the same experiment. In this case the accuracy of discrimination has increased to 96% for the three camera case as shown in Table 7. The increase in the accuracy indicate that the two sets of features capture different characteristics of an image, thereby enabling better identification of the source camera-model.

Table 7. The confusion table for 3 cameras corresponding to combined set of features

		Predicted		
		Nikon	Sony	Canon
Actual	Nikon	94.8%	1.5%	3.7%
	Sony	2.1%	95.3%	2.6%
	Canon	0%	2.3%	97.7%

6. Conclusions

In this paper, we attempt improve our previous approach to source camera-model identification problem. To detect traces of color interpolation (artifacts) in the RGB color channels, we incorporate a number of features tuned to capture the periodicity in the second-order derivatives with the features obtained through using EM algorithm 2. A classifier is then designed using the combined set of features and tested to determine the reliability of the selected features in discriminating the source camera-model among two and three cameras. This method is, limited to images that are not heavily compressed as the compression artifacts suppress and remove the spatial correlation between the pixels due to CFA interpolation.

References

[1] M. Kharrazi, H. T. Sencar and N. Memon, Digital Camera Model Identification, *Proceedings of the IEEE International Conference on Image Processing*, 2004.

[2] S. Bayram, H. T. Sencar and N. Memon, Source Camera-Model Identification Based on CFA Interpolation, *Proceedings of the IEEE International Conference on Image Processing*, 2005.

[3] Special Issue on Data Hiding, *IEEE Transactions on Signal Processing*, Vol. 41, No. 6, 2003.

[4] I. Avcibas, N. Memon and B. Sankur, Steganalysis using Image Quality Metrics, *IEEE Transactions on Image Processing*, Vol. 12, No. 2, 2003.

[5] S. Lyu and H. Farid, Detecting Hidden Messages Using Higher-Order Statistics and Support Vector Machines, *Proceedings of the Information Hiding Workshop*, 2002.

[6] J. Lukas, J. Fridrich and M. Goljan, Determining Digital Image Origin Using Sensor Imperfections, *Proceedings of the IS&T SPIE*, Vol. 5680, 2005.

[7] A. Popescu and H. Farid, Exposing Digital Forgeries by Detecting Traces of Re-sampling, *IEEE Transactions on Signal Processing*, Vol. 53, No. 2, 2005.

[8] A. C. Gallagher, Detection of Linear and Cubic Interpolation in JPEG Compressed Images, *Proceedings of the 2nd Canadian Conference on Computer and Robot Vision (CRV'05)*, 2005.

[9] J. Adams, K. Parulski and K. Sapulding, Color Processing in Digital Cameras, *IEEE IEEE Micro*, Vol. 18, No. 6, 1998.

[10] T. Moon, The Expectation Maximization Algorithm, *IEEE Signal Processing Magazine*, Vol. 13, Nov., 1996.

[11] C. Chang and C. Lin, LIBSVM: A library for support vector machines, 2001, Software available at http://www.csie.ntu.edu.tw/čjlin/libsvm.

DIGITAL IMAGE FORENSICS FOR IDENTIFYING COMPUTER GENERATED AND DIGITAL CAMERA IMAGES

Sintayehu Dehnie, (Taha Sencar, Nasir Memon)†*

Polytechnic University
5 MetroTech Center
Brooklyn, NY 11201

ABSTRACT

We describe a digital image forensics technique to distinguish images captured by a digital camera from computer generated images. Our approach is based on the fact that image acquisition in a digital camera is fundamentally different from the generative algorithms deployed by computer generated imagery. This difference is captured in terms of the properties of the residual image (*pattern noise* in case of digital camera images) extracted by a wavelet based denoising filter. In [1], it is established that each digital camera has a unique pattern noise associated with itself. In addition, our results indicate that the two type of residuals obtained from different digital camera images and computer generated images exhibit some common characteristics that is not present in the other type of images. This can be attributed to fundamental differences in the image generation processes that yield the two types of images. Our results are based on images generated by the Maya and 3D Studio Max software, and various digital camera images.

Index Terms: Computer graphics, image analysis, image classification, image processing, .

1. INTRODUCTION

Advances in digital imaging technologies raised new issues and challenges concerning the integrity and authenticity of digital images. Digital images can now be easily created, edited and manipulated without leaving any obvious traces of such operations. These capabilities undermine the credibility of digital images in all aspects. Digital image forensics is an emerging research field aiming at determining the origin and potential authenticity of a digital image.

One of the fundamental problems digital image forensics techniques attempt to solve is the identification of the source of a digital image. That is, to determine by what means a digital image has been created, e.g., digital camera, scanner, generative algorithms, etc. Possible solutions to the problem of image source identification may include one of the below approaches:

1. Verifying and evaluating the image statistics that are inherent to real-life sceneries and objects.
2. Detecting, classifying and measuring the qualities of spatial structures (i.e., color, texture and edge structures) in an image.
3. Identifying signatures to detect traces of certain types of operations used in image generation process by possible sources.

*Electrical and Computer Engineering Department
†Computer and Information Science Department

In this work, we study a specific instance of this problem which involves identifying whether a given image is a depiction of a real-life occurrence (and objects) or a fictitious realization. That is, distinguishing digital images generated by a digital camera from the ones generated by a computer graphics renderer. Our approach is motivated by the hypothesis that image acquisition in a digital camera includes many common processing stages (regardless of the specific digital camera used in capturing the image) leaving a unique signature in certain properties of the resulting image which may not necessarily be present in synthetically generated images. This is because the methodology governing the generative algorithms is fundamentally different from the image acquisition pipeline in a digital camera. Although this approach by itself cannot fully address the source identification problem (as it cannot resolve cases where a digital camera is used to capture the image of computer generated scene and objects), it is an important component of image forensics techniques.

In [1], Lukas et al. argued that images from a given digital camera exhibit a unique stochastic characteristic due to the pattern noise introduced in the medium to high frequency content of an image during image acquisition. Furthermore, they showed that the presence of the pattern noise can be detected by correlative processing, and an image can be uniquely associated with a digital camera through the known *reference error pattern*. In their work, the *reference error pattern* of a specific digital camera is the averaged noise pattern, obtained through image denoising, from a number of images captured by that camera. *In this paper, we exploit the fact that, although each individual camera has a unique noise pattern associated with it, pattern noise introduced by different digital cameras may have common (statistical) properties, as the deployed image sensor technology remains same, and that this common characteristic will not be present in computer generated images. Similarly, computer generated imagery may exhibit certain common properties, due to the use of same generative algorithms, that are not shared by the digital camera images.* Based on this argument, we investigate the potential of distinguishing computer generated images from digital camera images.

2. METHODOLOGY

As discussed in [1], for a given digital camera, the pattern noise remains approximately unchanged (regardless of the captured illumination from the scene) in each image, and it can be modeled as an additive noise. Furthermore, it is known that the pattern noise is relatively stable over the camera life span and a reasonable range of conditions such as temperature. Because of these properties we assume that traces of pattern noise is a reliable indicator that can be

used to distinguish digital camera images from computer generated images.

To test the validity of the assumption that digital camera and computer generated images are the result of two fundamentally different set of operations and that common properties of pattern noise associated with each type of images is not shared by the other type, we take an approach similar to that of [1]. For this purpose, we generate a reference noise pattern for a class of computer generated images using a given algorithm. We obtain the reference pattern by applying a wavelet based denoising filter [2] to extract the noise from each image. The denoising filter is derived from a bivariate statistical model that takes into account the statistical dependency between adjacent wavelet coefficients of natural images. This form of denoising filter is one of the best filters available for image denoising in the literature. Figure 1 shows the system block diagram. The denoising filter is locally adaptive[3] and includes a robust median estimator [4] in order to estimate the noise variance. Let X denote an image and $\hat{X}$ denote its denoised version. The pattern noise, e, is given by

$$e = X - \hat{X} \tag{1}$$

The reference noise pattern, e_{ref}, is obtained by averaging over

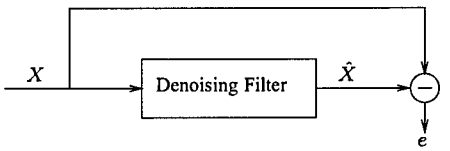


Fig. 1. System Model.

many instances of e.

The identification of image type is established by correlating the image residual with the pre-computed reference error pattern associated with a generative algorithm. To classify a given image X as digital camera or computer generated image, the normalized correlation between the residual image, e (1), and the reference error pattern of a generative algorithm is computed as

$$\rho = \frac{(e - E[e])(E_{ref} - E[E_{ref}])}{||e - E[e]||||E_{ref} - E[E_{ref}]||} \tag{2}$$

where $E[\,]$ is the expected value.

3. RESULTS

In our experiments we consider two sets of computer generated digital images. The first set is generated using Maya software, whereas the other set is generated using 3D Studio Max software. The images were obtained from publicly available websites [5] where it is explicitly noted that the images were generated by Maya and 3D Studio Max software, and other software suites, like Photoshop, for texture design purposes. The digital camera images are also obtained form publicly available websites [6] and divided into three sets. The first two sets are from a personal folder and involve images taken by two different cameras. The first set of images are taken by the (same) FUJI FinePixS2 Pro Digital Camera, whereas the second set of images are taken by the Kodak DCS Pro SLR/n Digital Camera. The third set involves images (each) taken by different digital cameras (from different folders) including various digital camera makes and models. In all cases, some of the images are used for obtaining the reference error pattern and the rest is used for test and evaluation of the method.

To establish the presence of a statistical difference between computer generated and digital camera images, we measured the statistics of each residual image for four different pairs of image sets. Each pair contains two sets of 100 Maya and digital camera images. The results are shown in Figure 2 and 3. We observed that the mean value of extracted noise from camera images is relatively higher. It is also observed that noise extracted from Maya images exhibit a relatively lower skewness (higher kurtosis).

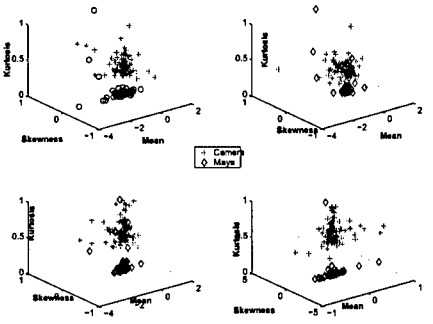


Fig. 2. Measured statistics of residual image for different sets of images.

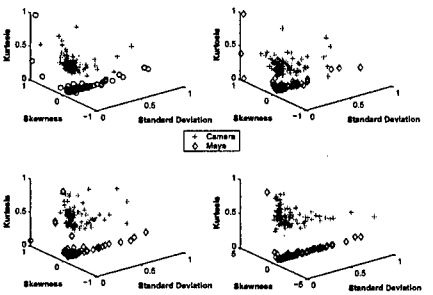


Fig. 3. Measured statistics of residual image for different sets of images.

We showed above the statistical variations in camera and Maya images. This statistical variation should be preserved in the reference error pattern generated from multiple images. In the next experiment we computed correlation of the reference error pattern with error extracted from test images. In our experiments, correlation of image residual with reference patterns is considered for three different cases. In the first case, the reference pattern is generated considering all subbands (HL, LH, and HH) in the wavelet transform domain. In the second case, the reference error pattern is generated by excluding the HL subband. In the last case, only the HH subband is considered when the reference error pattern is generated. In this paper we show results from the experiment that involves all the high-frequency wavelet coefficients. Figures 3 shows the correlation of the test images with the reference error pattern each obtained from the 300 images taken by different cameras. The figures depict that the test camera images exhibit stronger correlation with the reference error pattern. On the other hand, we observed that Maya and 3D

Studio Max test images have weaker but non-zero correlation to the reference error pattern. The statistics (histogram) for the computed correlation coefficient are as shown in Figure 4. The mean values for the computed correlation are (0.0319, 0.0395) for Maya and 3D Studio Max test images respectively. On the other hand, the mean of the correlation of the camera test image is 0.1228. This further validates the argument that digital cameras have common stochastic features that may not be present in computer generated images. To measure

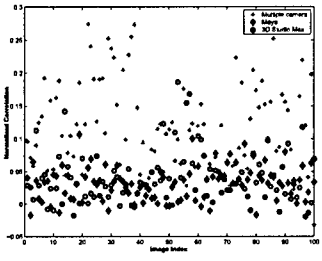
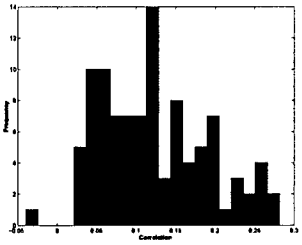
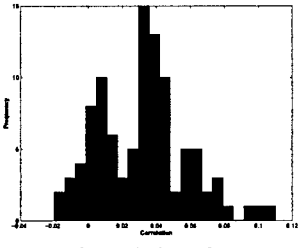


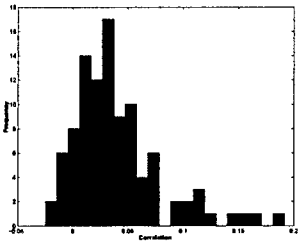
Fig. 4. Correlation of test image residual with reference error pattern obtained from different cameras.



(a) Histogram of correlation of camera test image.



(b) Histogram of correlation of Maya test image.



(c) Histogram of correlation of 3D Studio Max test image.

Fig. 5. Correlation statistics of test image residual with multi-camera reference error pattern.

the false positive rate of the above experiment, ROC curves were generated and are shown in Figure 5. We repeated the experiment

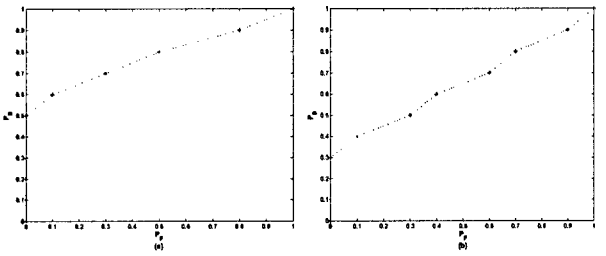


Fig. 6. ROC curves to measure false positive rate, (a) Maya, (b) 3D Studio Max.

using reference error patterns obtained from the 300 images taken by FUJIFinePix S2 Pro and Kodak DCS Pro SLR/n respectively. Figures 6 and 7 show the corresponding correlation for each reference error pattern with the test images. Parallel to the results in [1], we observed that test images (from the same camera) showed stronger correlation with the reference error pattern. Similar to the previous experiment, Maya and 3D Studio Max images exhibit weaker but non-zero correlation. We did a similar experiment by obtaining a

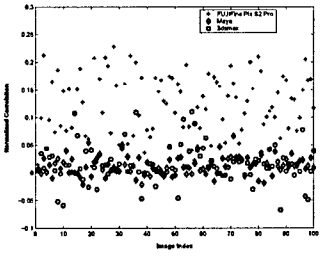


Fig. 7. Correlation of test image residual with camera reference error pattern.

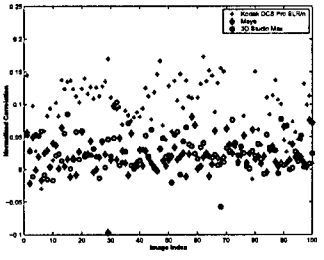


Fig. 8. Correlation of test image residual with camera reference error pattern.

Maya reference error pattern (e_{maya}) using 300 images. The results are shown in Figure 8. Similar to the previous two cases, a stronger correlation of the Maya reference error pattern with the test Maya images is observed. In agreement with our argument, the correlation of the camera test images with the Maya reference pattern exhibit weaker but non-zero correlation. This is an indication that Maya images also have a unique stochastic feature. The correlation statistics is as shown in Figure 9. We also measure the false positive rate of the above experiment by generating ROC curves as is shown in Figure 10. We repeated the experiment using Maya reference error

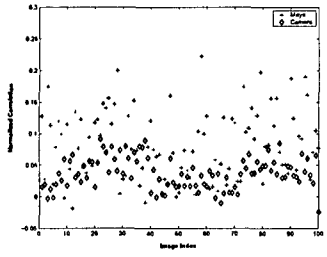


Fig. 9. Correlation of test image (different cameras) residual with Maya reference error pattern.

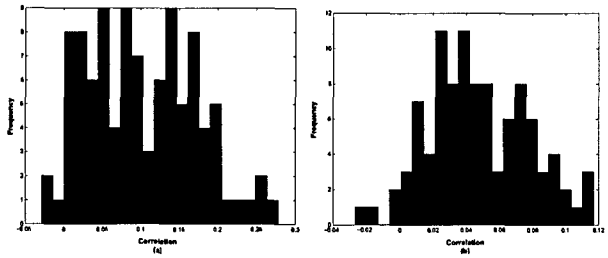


Fig. 10. Statistics of (histogram) correlation with Maya reference error pattern, (a) Maya, (b) Cameras

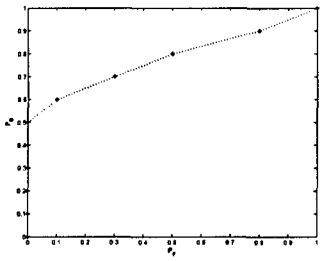


Fig. 11. ROC curve to measure false positive rate.

pattern and two sets of test camera obtained from two different cameras. The results are shown in Figure 10. Ultimately, to verify that

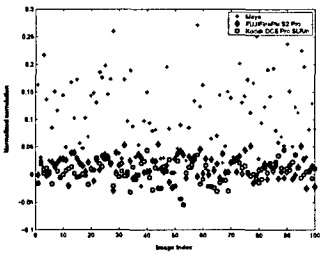


Fig. 12. Correlation of test images with Maya reference pattern.

properties of Maya reference pattern is not image set dependent, we generated two (non-intersecting) sets of 150 Maya images. We obtained the reference error pattern and computed the correlation with the test Maya images. Figure 12 depicts the measured correlation. These values establish consistency for Maya reference error pattern in capturing the common properties of the residual error in Maya images.

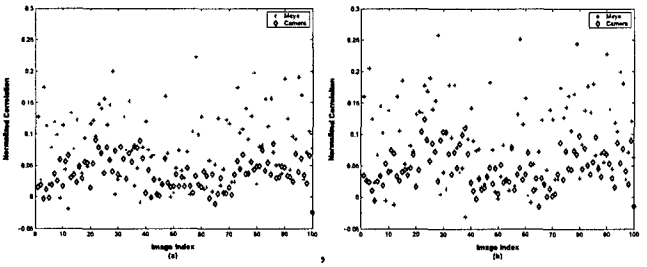


Fig. 13. Correlation of test image correlation with two (different) Maya reference error pattern, (a) Set I, (b) Set II.

4. CONCLUSION

In this paper, we argued that digital camera images exhibit a common statistical property which is not present in computer generated images and vice versa. Based on this argument, we proposed a method to differentiate between digital camera images from computer generated images.

We observed that test Maya images exhibit higher correlation with the Maya reference error pattern. The higher correlation in Maya images indicates the presence of unique statistical properties in Maya images.On the other hand, we observed low correlation with the Maya reference error pattern when test images from a given camera and multiple camera are used. We also showed consistency in Maya reference error pattern using two sets of Maya images. Mixed test images exhibit relatively higher correlation with the considered digital camera reference error pattern compared to Maya test images. This further validates the argument that digital camera images exhibit common statistical properties.

5. REFERENCES

[1] Jan Lukas, Jessica Fridrich, and Miroslav Goljan, "Determining digital image origin using sensor imperfections," in *SPIE Electronic Imaging*, January 16-20 2005, pp. 249–260.

[2] Levent Sendur and Ivan Selesnick, "Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency," *IEEE Transaction on Signal Processing*, vol. vol. 50, no. no. 11, November 2002.

[3] Levent Sendur and Ivan Selesnick, "Bivariate shrinkage with local variance estimation," in *Signal Processing Letters*, vol. 9. IEEE, December 2002.

[4] D.L. Donoho and I.M. Johnstone, "Ideal spatial adaptation by wavelet shrikage," *Biometrika*, vol. 81, no. 3, pp. 425–455, 1994.

[5] "www.3dlinks.com,www.alias.com, www.highend3d.com," .

[6] "www.freefoto.com and www.dpreview.com," .

[7] Tian-Tsong Ng, Shih-Fu Chang, Jessie Hsu, Lexing Xie, and Mao-Pei Tsui, "Physics-motivated features for distinguishing photographic images and computer graphics," in *Proceedings of the 13th annual ACM international conference on Multimedia*, Singapore, November 2005, pp. 239–248, ACM Press.

SOURCE CAMERA IDENTIFICATION BASED ON SENSOR DUST CHARACTERISTICS

A. Emir Dirik

Polytechnic University
Department of Electrical
and Computer Engineering
Brooklyn, NY, US

*Husrev T. Sencar, Nasir Memon**

Polytechnic University
Department of Computer
and Information Science
Brooklyn, NY, US

ABSTRACT

A problem associated with digital single lens (DSLR) cameras is *sensor dust*. This problem arises due to dust particles attracted to the sensor, when the interchangeable lens is removed, creating a dust pattern in front of the imaging sensor. Sensor dust patterns reveals themselves as artifacts on the captured images and they become more visible at smaller aperture values. Since this pattern is not changed unless the sensor surface is cleaned, it can be used to match a given image to source DSLR camera. In this paper, we propose a new source camera identification method based on sensor dust characteristics. Dust specks on the image are detected using intensity variations and shape features to form the dust pattern of the DSLR camera. Experimental results show that the method can be used to identify the source camera of an image at very low false positive rates.

1. INTRODUCTION

In today's digital age, the creation and manipulation of digital images is made simple by digital processing tools that are easily and widely available. As a consequence, we can no longer take the authenticity of digital images for granted. Today, there is a severe lack of techniques and methodologies for verifying the integrity of digital images. Due to this asymmetry, digital images appear to be the source of a new set of problems. This is especially true when it comes to legal photographic evidence. Image forensics, in this context, is concerned with uncovering some underlying fact about an image. To address these problems, more recently, several digital image forensics techniques have been proposed for both image forgery detection [1, 2, 3, 4, 5] and image source identification [6, 7, 8, 9, 10, 11, 12].

In image source identification problem, one of the most pressing concerns is the ability to match an image to its source camera. In this context, the most promising approach is proposed by Lukáš, et. al. [12]. In their method, sensor's pattern noise is used to identify the source of an image. Sensor pattern noise is caused by various factors, such as dust specks on optics, interference in optical elements, dark currents, etc. However, the high frequency component of the pattern noise can be modeled as additive noise and estimated by applying a wavelet based denoising to the captured image. Then, the extracted noise residues from multiple images are averaged to estimate the camera's noise pattern, i.e., reference pattern. To identify the source of a given image, the noise residue of the image in question is correlated with the reference noise patterns extracted from the camera.

In this paper, a new method based on sensor dust characteristics of DSLR cameras for image source identification is proposed. Essentially, the lenses on DSLR cameras are interchangeable and the sensor dust problem arises when the interchangeable lens is removed, thereby opening the sensor area to the hazards of dust and moisture. Once the lens is taken off, the dust particles around the camera are attracted to the imaging sensor by electrostatic fields resulting a dust pattern on the surface of the sensor. (It should be noted that, the dust isn't actually sitting on the sensor itself, but on the element just in front of it. These elements include the dichroic mirror or low-pass filter.) This dust pattern can be seen as small specks, in the form of localized intensity degradations, all over the image under some certain conditions, especially with small aperture settings. In figure 1 a sample image[1] taken with DSLR camera with dust specks are shown. Although it is very hard to locate dust positions, when blockwise local histogram equalization is applied to each pixel in the image, sensor dust artifacts can be easily seen.

Another aspect of the problem is that sensor dust is cumulative. That is, with every change of the lens, more dust is likely to be added to sensor, thereby worsening the problem over the time. Furthermore, most state-of-the-art digital cameras do not offer a built-in solution for removal of sensor dust. On the other hand, the process of sensor cleaning, through swabbing, brushing, using compressed air, brings with it the risk of scratching the sensor. Therefore, sensor dust is a persistent problem that appears to be getting widespread with the

*This work is supported by National Institute of Justice grant 2006-92251-NY-IJ.

[1] image is downloaded from www.pbase.com

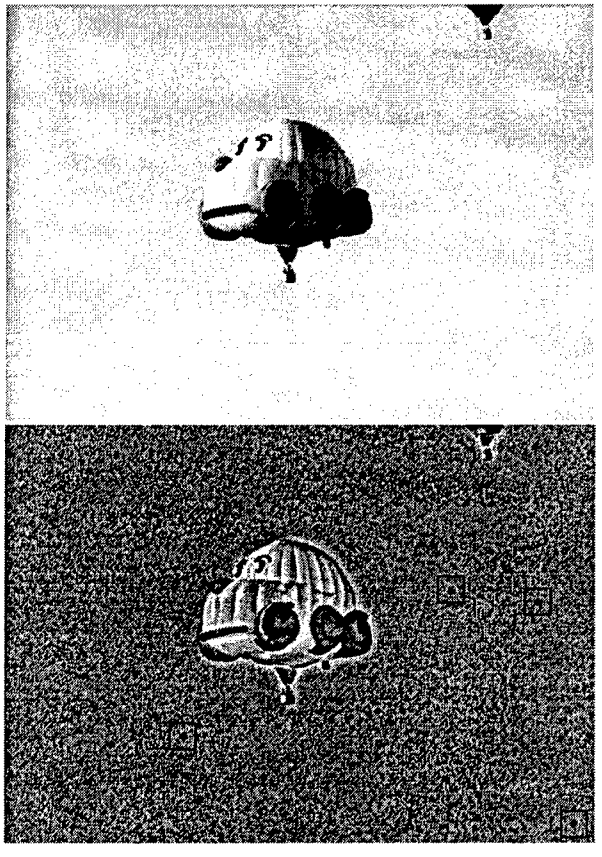


Fig. 1. Sample image taken by NikonD70, f-number:F/14 (up), local histogram equalization result (down). The black boxes show the location of dust specks in the image

advent of DSLR cameras due to superior image quality they provide. It should be noted that since sensor dust problem is not intrinsic to cheaper consumer cameras, the detection of any sensor dust in a given image can be evaluated as a proof of the image source being a DSLR camera. Moreover, with the knowledge of dust positions/pattern in a given image and camera, it is possible to associate images with a particular DSLR camera.

In the following sections a method to locate dust specks in a given image is described. This is primarily achieved by comparing the dust positions of a given image with those of the particular DSLR sensor dust pattern. The efficacy of the proposed method is substantiated by experimental results.

2. SENSOR DUST CHARACTERISTICS AND THEIR FORENSIC USE

Sensor dusts reveal themselves in photos taken with smaller apertures settings and they become less noticeable with increasing aperture values. This is due to the fact that dust spots stand a distance from the actual sensor and wide aperture values let more light to go around the dust spots. Hence the shadow of the dust (speck) on the color sensor shows up in the image as a blurry, soft speck. On the contrary, at small aperture values, the light source can be assumed to be a small pinpoint spotlight as a result of which specks become dark and hard edged [13, 14]. In Fig. 2, the dust spots for two different aperture settings, f/22 and f/32, are shown. It can be seen that the change in f-number affects the intensity and radius of the dust speck and with the increase in f-number (aperture gets smaller) the dust speck gets more darker and smaller.

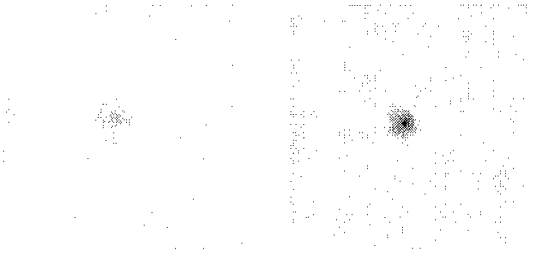


Fig. 2. Dust specks with different apertures, f/22(left), f/32(right)

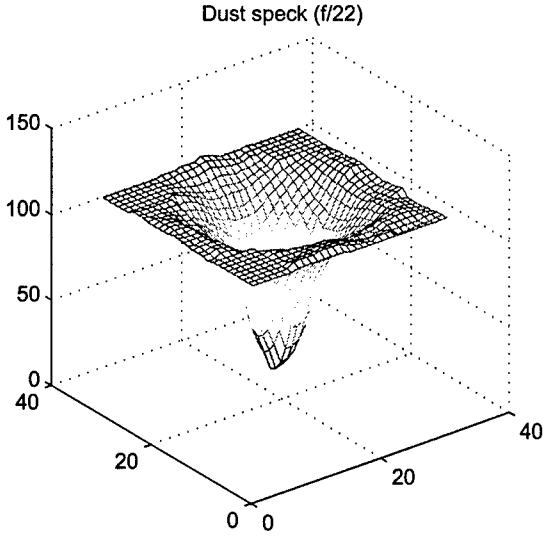


Fig. 3. Intensity loss due to dust speck (f/22)

To exploit this vulnerability of DSLR cameras, we detect traces of sensor dust in images and use it for source identifi-

cation. For this purpose, we initially aim at determining the presence of dust specks on an image. Due to difficulty in discriminating effects of dust on images from the image content, specifically in the textured and parts with high frequency content, the core element of the method is the dust detection. In other words, the crux of the method lies in dust modeling which essentially determines the rate of false-positives–a crucial parameter concerning its forensic use.

Although some DSLR cameras have anti-dust mechanisms, they can not keep the sensor surface clean completely. Some camera manufacturers also provide post-processing tools to remove dust specks on images based on dust template photos taken with high aperture settings. In the market there are also a couple of commercial softwares which detect and remove dust traces from a single image. Nevertheless all these tools have high false positives. There are also several patents for dust speck detection and removal [14, 15, 16]. In [14] local intensity variations in uniform regions are assumed as dust spots. In [15], likely dust specks are detected by taking the second order derivative of the image and the peaks of the derivatives are assigned as dust positions. In [16] dust positions are detected by taking first-order-derivative and applying some post-processing operations. However, our initial experimental studies showed that gradient based dust spec detection methods suffer from relatively high error rates (miss and false positive probabilities). Therefore, in this work, we did not consider to use any gradient based search method to locate dust positions. Our sensor dust detection method is described below.

2.1. Dust model

Our sensor dust model relies on the observation that sensor dust has two major characteristics: (a) causing an abrupt change on the intensity surface (e.g., intensity loss) depending on the aperture size; and (b) appearing most generally the form of rounded shapes, see figures 2 and 3. To model the intensity degradation due to sensor dust we utilize a 2D inverse gaussian function with a particular standard deviation and gain. It should be noted that as f-number increases the diameter of the dust spot in the image decreases and the intensity loss in the dust spot increases. Moreover, the shape of the intensity loss becomes more kurtotic. Since the dimensions of the dust is related with aperture, its is also essential to detect f-number to locate dust specks properly. (In our work we assume the EXIF data of the image is not available.)

To locate the position of the dust speck, we apply fast normalized cross-correlation [17] with estimated dust model as in equation 1.

$$dustmodel(x,y) = -G\frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{x^2+y^2}{2\sigma^2}} \qquad (1)$$

In the equation G refers to the intensity loss. The diameter of the dust speck is controlled by σ. We estimate the diameter

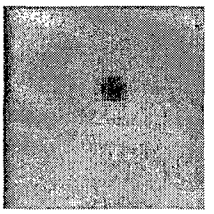


Fig. 4. Dust speck

of dust specks based on cross-correlation results obtained under different σ values ranging from 1 to 3. The sigma value which produces the maximum cross-correlation is chosen as the dust model parameter, and the corresponding correlation output is used to detect dust specks. Once the correlation output is computed, the local maximums higher than a empirically determined threshold (such as 0.4) are labeled as dust candidates. In order to eliminate false positives, dust candidates in highly detailed regions are ignored.

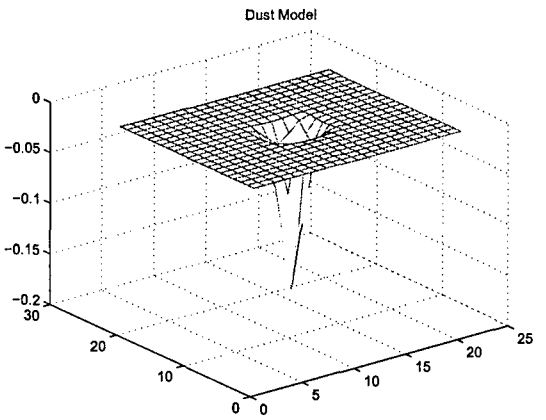


Fig. 5. Dust model, G=1.0, $\sigma = 1.0$

2.2. Contour analysis

Although cross-correlation method works well in smooth regions, it may produce high correlations on edges and textured regions. In order to reduce this sort of false positives, we apply further analysis on each dust candidate based on their local contour characteristics. For each dust candidate we compute their contour map as shown in Fig. 7. Apart from the correlation output, we locate the dust center by analyzing the local minimums which have maximum number of closed loops around. Then, the intensity loss in the possible dust speck region and the eccentricity of the dust contour, which indicates how contour shape resembles to a circle, are computed. These parameters then combined together to compute a normalized

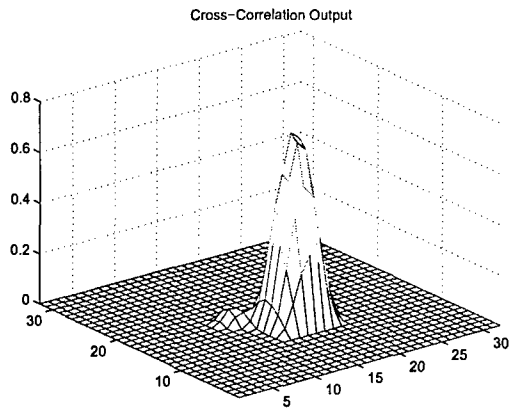


Fig. 6. Cross-correlation output of dust speck in fig. 4

confidence value of the contour region. If there is not any significant intensity loss inside of the contour plot then the confidence value is assigned to zero. After contour analysis, according to the confidence values, each candidate is evaluated to determine the dust specs.

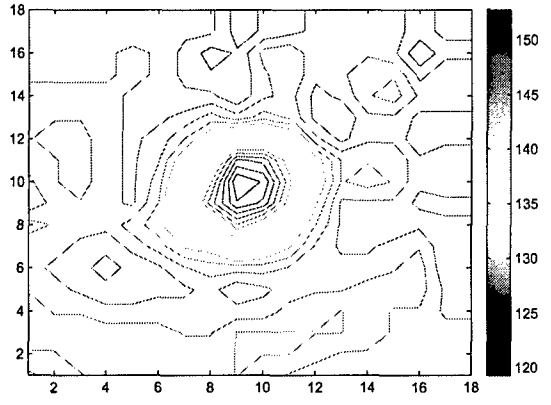


Fig. 7. Contour analysis of dust speck in fig. 4 (Num. of closed loops inside of the speck : 16, intensity loss : 24.3)

2.3. Camera Dust Pattern Generation

To be able to address a forensic setting we assumed two relevant scenarios of dust pattern generation.

- Digital camera is available: In this case the dust pattern of an image can be generated by taking the picture of distant smoothly varying scenery by manually setting the focal length to high values (f/32 or f/36). Then proposed dust detection method is applied to create dust pattern of the camera.
- Images acquired with the DSLR camera are available: When the camera is not available but rather a number of images taken by the camera is present, the dust points that are determined by correlation and through shape characteristics in each image is superimposed together to form the dust pattern/template of the camera. Once the template is created, a threshold is applied to the template to reduce the number of falsely labeled specks in the dust pattern. The underlying idea of applying a threshold to the template is that the actual dust specks should show up at least in two or more images. Since the probability of getting a false dust candidate at the same position in multiple images is very low, we expect that false positives due to image content will be eliminated after thresholding. The dust candidates which have higher confidence values than a fixed threshold are considered to represent the dust pattern of the camera.

Finally, source camera identification model is realized by matching camera dust template with the estimated dust pattern of a given image

3. EXPERIMENTAL RESULTS

Our experiments are based on the assumption that the digital camera is not available and that the sensor dust pattern has to be obtained from a number of images taken by a DSLR camera where obtaining a precise dust template is not easy. To create an image set we have downloaded DSLR images from three different personal galleries at `www.pbase.com`. All images are taken with Sigma SD10. We also created an alternative image data set taken from compact consumer cameras. In order to reduce computation time of cross-correlation, all images are resized to 800x533 pixels. Since dust spots are almost invisible at large aperture rates, images with low f-numbers (below than 8) are not used at experiments.

From each three gallery, we randomly select 10 images to create a dust template. As described in Section 2, we computed the cross-correlation outputs for each image and then superimposed all the outputs to create a camera dust pattern. The contour analysis is then used to refine the final result. After dust patterns of three cameras are computed, in the testing and verification step previously unseen images in each image gallery are analyzed to determine if they include any traces of dust patterns in the locations pointed in the dust template of the camera.

In Figures 8,9,10 we provide results, when the dust template is generated only from 10 images, and tested on 20 images taken by the same and 60 random images taken by other cameras. Our matching results indicate that, we achieve a detection rate around 92% with 0% false positive rate by setting the confidence threshold 1.2. In the figures x-axes shows

the image index and y-axes is the proposed metric indicating confidence in the match.

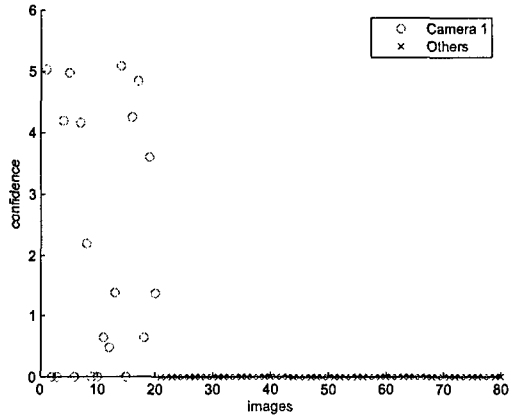


Fig. 8. Num. of matches between the template of the camera 1 and dust candidates. (num. of dusts in the template : 15)

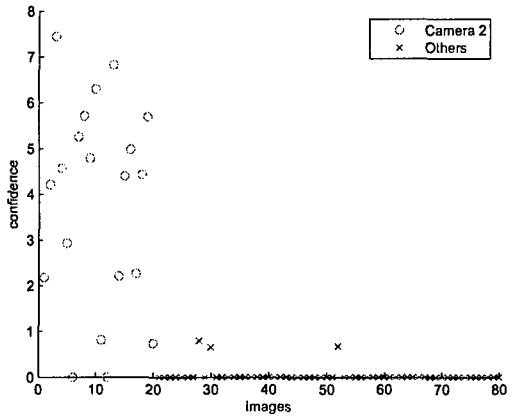


Fig. 9. Num. of matches between the template of the camera 2 and dust candidates. (num. of dusts in the template : 16)

4. DISCUSSION

In this work, we present a source camera identification model for images taken from DSLR cameras. We show that it is possible to associate a given image with a particular DSLR camera with very low false alarm rates using sensor dust characteristics. Though we tested our model with a small set of DSLR cameras, our experimental results are promising. However, there are some problems inherent to the proposed approach. The most important one is that for wide apertures dust specks become almost invisible and detection of the dust

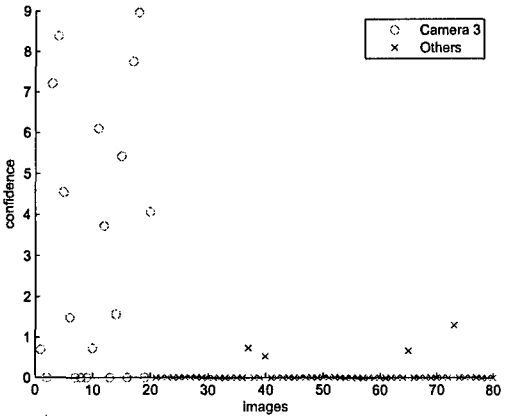


Fig. 10. Num. of matches between the template of the camera 3 and dust candidates. (num. of dusts in the template : 38)

speck becomes a challenging task. Another important problem is the detection of dust specks in non-smooth, complex regions without yielding many false-positives. In the future work, we will address these issues.

5. REFERENCES

[1] A. Swaminathan, M. Wu, and K. J. Ray Liu, "Image tampering identification using blind deconvolution," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2006, pp. 2311–2314.

[2] S. Bayram, I. Avcibas, B. Sankur, and N. Memon, "Image manipulation detection," *Journal of Electronic Imaging*, vol. 4, October-December 2006.

[3] M. Johnson and H. Farid, "Exposing digital forgeries by detecting inconsistencies in lighting," in *ACM Multimedia and Security Workshop*, 2005.

[4] A. Popescu and H. Farid, "Exposing digital forgeries in color filter array interpolated images," *IEEE Transactions on Signal Processing*, vol. 53, no. 10, pp. 3948–3959, 2005.

[5] A. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of re-sampling," *IEEE Transactions on Signal Processing*, 2004.

[6] M. Kharrazi, H. T. Sencar, and N. Memon, "Blind source camera identification," in *IEEE International Conference on Image Processing*, October 2004, vol. 1, pp. 709–712.

[7] S. Bayram, H. T. Sencar, N. Memon, and I. Avcibas, "Source camera identification based on cfa interpolation," in *Proc. of IEEE ICIP*, 2005.

[8] A. Swaminathan, M. Wu, and K. J. Ray Liu, "Non-intrusive forensic analysis of visual sensors using output images," in *IEEE Conference on Acoustic, Speech and Signal Processing (ICASSP)*, France, May 2006, vol. 5, pp. 401–404.

[9] H. Farid and S. Lyu, "Higher-order wavelet statistics and their application to digital forensics," *Conference on Computer Vision and Pattern Recognition Workshop*, vol. 8, 2003.

[10] N. Tian-Tsong, C. Shih-Fu, H. Yu-Feng, X. Lexing, and T. Mao-Pei, "Physics-motivated features for distinguishing photographic images and computer graphics," in *ACM Multimedia*, Singapore, November 2005.

[11] Y. Wang and P. Moulin, "On discrimination between photorealistic and photographic images," in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, May 2006, vol. 2, pp. II–161– II–164.

[12] J. Lukas, J. Fridrich, and M. Goljan, "Digital camera identification from sensor noise," *IEEE Transactions on Information Security and Forensics*, vol. 1, pp. 205–214, June 2006.

[13] R. Willson, M. Maimone, A. Johnson, and L. Scherr, "An optical model for image artifacts produced by dust particles on lenses," in *8th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, September 2005.

[14] E. Steinberg, Y. Prilutsky, P. Corcoran, and et al., "Method of detecting and correcting dust in digital images based on aura and shadow region analysis," United States Patent Application Publication, March 2005, US Patent, 0068448 A1.

[15] Y. Morimoto, "Image processing method, an image processing apparatus, and a storage medium readable by a computer," United States Patent, September 2000, Patent number, 6,125,213.

[16] A. Krainiouk and R. T. Minner, "Method and system for detecting and tagging dust and scratches in a digital image," United States Patent, May 2001, Patent number, 6,233,364 B1.

[17] J. Lewis, "Fast normalized cross-correlation," in *Proc. of Vision Interface*, 1995.

Journal of Electronic Imaging 15(4), 041102 (Oct–Dec 2006)

Image manipulation detection

Şevinç Bayram
İsmail Avcıbaş
Uludağ University
Department of Electronics Engineering
Bursa, Turkey

Bülent Sankur
Boğaziçi University
Department of Electrical and Electronics Engineering
Istanbul, Turkey

Nasir Memon
Polytechnic University
Department of Computer and Information Science
Brooklyn, New York
E-mail: sevincbayram@hotmail.com

Abstract. *Techniques and methodologies for validating the authenticity of digital images and testing for the presence of doctoring and manipulation operations on them has recently attracted attention. We review three categories of forensic features and discuss the design of classifiers between doctored and original images. The performance of classifiers with respect to selected controlled manipulations as well as to uncontrolled manipulations is analyzed. The tools for image manipulation detection are treated under feature fusion and decision fusion scenarios. © 2006 SPIE and IS&T.* [DOI: 10.1117/1.2401138]

1 Introduction

The sophisticated and low-cost tools of the digital age enable the creation and manipulation of digital images without leaving any perceptible traces. As a consequence, one can no longer take the authenticity of images for granted, especially when it comes to legal photographic evidence. Image forensics, in this context, is concerned with reconstructing the history of past manipulations and identifying the source and potential authenticity of a digital image. Manipulations on an image encompass processing operations such as scaling, rotation, brightness adjustment, blurring, contrast enhancement, etc. or any cascade combinations of them. Doctoring images also involves the pasting one part of an image onto another one, skillfully manipulated so to avoid any suspicion.

One effective tool for providing image authenticity and source information is digital watermarking.[1] An interesting proposal is the work of Blythe and Fridrich[2] for a secure digital camera, which losslessly embeds the photographer's iris image, the hash of the scene image, the date, the time, and other camera/picture information into the image of the scene. The embedded data can be extracted later to verify the image integrity, establish the image origin, and verify the image authenticity (identify the camera and the photographer). However, its use requires that a watermark be embedded during the creation of the digital object. This limits watermarking to applications where the digital object generation mechanisms have built-in watermarking capabilities. Therefore, in the absence of widespread adoption of digital watermarking technology (which is likely to continue for the foreseeable future), it is necessary to resort to image forensic techniques. Image forensics can, in principle, reconstitute the set of processing operations to which the image has been subjected. In turn, these techniques not only enable us to make statements about the origin and veracity of digital images, but also may give clues as to the nature of the manipulations that have been performed.

Several authors have recently addressed the image forensic issue. Popescu *et al.*[3] showed how resampling (e.g., scaling or rotating) introduces specific statistical correlation, and described a method to automatically detect correlations in any portion of the manipulated image. Avcibas *et al.*[4] developed a detection scheme for discriminating between "doctored" images and genuine ones based on training a classifier with image quality features, called "generalized moments." Both methods are, however, limited to a subset of doctoring operations. Johnson and Farid[5] described a technique for estimating the direction of an illuminating light source, based on the lighting differences that occur when combining images. Popescu and Farid[6] quantified the specific correlations introduced by color filter array (CFA) interpolation and described how these correlations, or lack thereof, can be automatically detected in any por-

Paper 06115SSR received Jun. 30, 2006; revised manuscript received Sep. 19, 2006; accepted for publication Sep. 21, 2006; published online Dec. 28, 2006.
1017-9909/2006/15(4)/041102/17/$22.00 © 2006 SPIE and IS&T.

(a) (b) (c)

Fig. 1 Example of photomontage image: the head of the child in the center of the group is replaced, after appropriate manipulation, with that of the child in the middle photograph.

tion of an image. Fridrich *et al.*[7] investigated the problem of detecting the copy-move forgery and proposed a reliable method to counter this manipulation.

The problem addressed in this paper is to detect doctoring in digital images. Doctoring typically involves multiple steps, which typically involve a sequence of elementary image-processing operations, such as scaling, rotation, contrast shift, smoothing, etc. Hence, to tackle the detection of doctoring effects, we first develop single tools (experts) to detect these elementary processing operations. Then we show how these individual "weak" detectors can be put together to determine the presence of doctoring in an expert fusion scheme. Novel aspects of our work in this paper are the following. First, we introduce and evaluate features based on the correlation between the bit planes as well the binary texture characteristics within the bit planes. These are called binary similarity measures (BSMs) as in Ref. 8. We compare their performance against two other categories of tools that were previously employed for image forensics and steganalysis, namely, IQMs (image quality measures)[9,10] and HOWS (higher order wavelet statistics).[11] Second, we unify these three categories of features, IQMs, BSMs, and HOWSs, in a feature scenario. The cooperation between these feature categories is attained via a feature selection scheme formed from the general pool. It is shown that the feature fusion outperforms classifier performance under individual category sets. Third, we conduct both controlled and uncontrolled experiments. The controlled experiments are carried on a set of test images using image-processing tools to give us insight into the feature selection and classifier design. An example of controlled experiment is the blurring of the whole image and its detection with a classifier that may be clairvoyant or blind. The uncontrolled experiments relate to photomontage images, where we cannot know specifically the manipulation tools used and where only parts of the image are modified with a cascade of tools.

An example of the photomontage effect is illustrated in Fig. 1, where the head of the central child in Fig. 1(a) is replaced with the head borrowed from the image in 1(b), after an appropriate set of manipulations such as cropping, scaling, rotation, brightness adjustment, and smoothing along boundaries. The resulting image is given in Fig. 1(c).

The organization of this paper is as follows: Section 2 reviews the forensic image features utilized in developing our classifiers or "forensic experts." Section 3 presents a detailed account of controlled and uncontrolled experiments. Conclusions are drawn in Sec. 4.

2 Forensic Features

Investigation of sophisticated manipulations in image forensics involves many subtleties because doctoring operations leave weak evidence. Furthermore, manipulations can be cleverly designed to eschew detection. Hence, an image must be probed in various ways, even redundantly, for detection and classification of doctoring. Furthermore, discriminating features can be easily overwhelmed by the variation in image content. In other words, the statistical differences due to image content variation can confound statistical fluctuations due to image manipulation. It is, thus, very desirable to obtain features that remain independent of the image content, so that they would reflect only the presence, if any, of image manipulations. The three categories of forensic features we considered are as follows:

1. *IQMs*. These focus on the difference between a doctored image and its original version. The original not being available, it is emulated via the blurred version of the test image. The blurring operation purportedly removes additive high-frequency disturbance due to certain types of image manipulations to create a version of the untampered image. The 22 IQMs considered in Refs. 9 and 10 range from block SNR to spectral phase and from spectral content to Spearman rank correlation.
2. *HOWS*. These are extracted from the multiscale decomposition of the image.[11] The image is first decomposed by separable quadrature mirror filters and the mean, variance, skewness, and kurtosis of the subband coefficients at each orientation and scale are computed. The number of HOWS features is 72.
3. *BSMs*. These measures capture the correlation and texture properties between and within the low-significance bit planes, which are more likely to be affected by manipulations.[8,12,13] The number of BSM features is 108.
4. *Joint Feature Set (JFS)*. We have considered the pooled set consisting of the three categories of features, namely the union of the IQM, BSM, and HOWS sets. This provides a large pool of features to choose from, that is, 108 BSMs, 72 HOWS, and 8 IQM features; overall, 188 features.
5. *Core Feature Set (CFS)*. We decided to create a core set, fixed in terms of the number and types of features, to meet the challenge of any potential manipulation. The motivation for this smaller core set of unique features was to avoid the laborious process of

feature selection for every new scenario. In other words, we envision a reduced set of features standing ready to be trained and used as the challenge of a new manipulation scenario arises. Obviously the performance of the CFS, which can only for classifier weights, would be inferior to the performance of the JFS, which can both chose features and train classifier weights. The common core of features was extracted as follows. We select the first feature from the set of 188 available features, as the one that results in the smallest average error of the semiblind classifiers (defined later); the second one is selected out of remaining $188-1$ features, which, as a twosome feature, results in the smallest average classification error, and so forth.

The feature selection process was implemented with the sequential forward floating search, (SFFS) method.[14] The SFFS method analyzes the features in ensembles and can eliminate redundant ones. Pudil *et al.*[14] claims that the best feature set is constructed by adding to and/or removing from the current set of features until no more performance improvement is possible. The SFFS procedure can be described as follows:

1. Choose from the set of K features the best two features; i.e., the pair yielding the best classification result.
2. Add the most significant feature from those remaining, where the selection is made on the basis of the feature that contributes most to the classification result when all are considered together.
3. Determine the least significant feature from the selected set by conditionally removing features one by one, while checking to see if the removal of any one improves or reduces the classification result. If it improves, remove this feature and go to step 3, otherwise do not remove this feature and go to step 2.
4. Stop when the number of selected features equals the number of features required.

The SFFS was run for each type of image manipulation, for the category of manipulations, for the pooled categories. The litmus test for feature selection was the performance of the regression classifier.[15] To preclude overtraining the classifier, we upper bounded the number of features selected by 20. This means that, e.g., at most 20 BSM features could be selected from the 108 features in this category. On the other hand, for the joint set and for the core set, the upper bound of feature population was set to 30. Often, however, the selection procedure terminated before the upper bound was reached.

We used the following definitions of classifiers:

1. *Clairvoyant classifier.* This is the classifier trained for a specific manipulation at a known strength. For example, we want to distinguish pristine images from the blurred ones, where the size of the blurring function aperture was n pixels. Obviously, this case, where one is expected to know both the manipulation type and its parameters is somewhat unrealistic in practice, but it is otherwise useful for understanding the detector behavior.
2. *Semiblind classifier.* This is the classifier for a specific manipulation at unknown strength. For example, we want to determine whether an image has been blurred after its original capture, whatever the blur size.
3. *Blind classifier.* This is the most realistic classifier if one wants to verify whether or not an image has been tampered with. For example, given an image downloaded from the Internet, one may suspect that it might have been manipulated, but obviously one cannot know the type(s) of manipulations.

To motivate the search for forensic evidence, we illustrate the last three bit planes of the "Lena" image, when the latter was subjected to blurring, scaling, rotation, sharpening, and brightness adjustment, as shown in Figs. 2 and 3. As shown in these examples, image manipulations alter to varying degrees the local patterns in bit planes (Fig. 2) as well as across bit planes (Fig. 3). Consequently, the statistical features extracted from the image bit planes can be instrumental in revealing the presence of image manipulations. Since each bit plane is also a binary image, it is natural to consider BSMs as forensic clues. BSMs were previously employed in the context of image steganalysis.[10,12] Similarly, these disturbance patterns will affect the wavelet decomposition of the image and the predictability across bands, which can be captured by HOWS. Finally, a denoising operation on the image will remove the content but will bring forth patterns similar to those shown in Figs. 2 and 3. Features trained to distinguish these patterns take place in the repertoire of IQMs.

Figure 4 illustrates the behavior of three selected features, each from one category, vis-à-vis the strength of manipulation. The continuous dependence of sample measures, the Sneath and Sokal[16] measure of BSMs; the normalized correlation measure of IQMs; and variance of vertical subband of HOWS (all to be defined in the following subsections) on the strength parameter for three types of manipulation is revealing.

2.1 *BSMs*

BMSs for images and their steganographic role were discussed in Refs. 4, 12, and 17 and Appendix A details concerning them. We conjecture that they can play an effective role in the evaluation of doctored images. Consider for example, the Ojala histogram as one of the BSM features. Figure 5(a) shows in the left column the 256-level gray-level histograms of the "Lena" image side by side with the 512-level Ojala histograms in the right column. The first row contains the respective histograms of the original images, while the following rows show the respective histograms in the manipulated images. Notice that while the gray-level histograms remain unperturbed, the Ojala histograms are quite responsive to the type of manipulation. For example, sharpening flattens the histogram, while rotation and blurring causes the probability of certain patterns to peak. The sensitivity of the Ojala histograms to manipulations can be quantified in terms of distance functions. In this paper, bit plane pairs 3-4, 4-5, 5-6, 6-7, and 7-8 for the red channel and bit plane pair 5-5 of the red and blue channels were used; in other words, the BSM features from these plane pairs were offered to the feature selector.

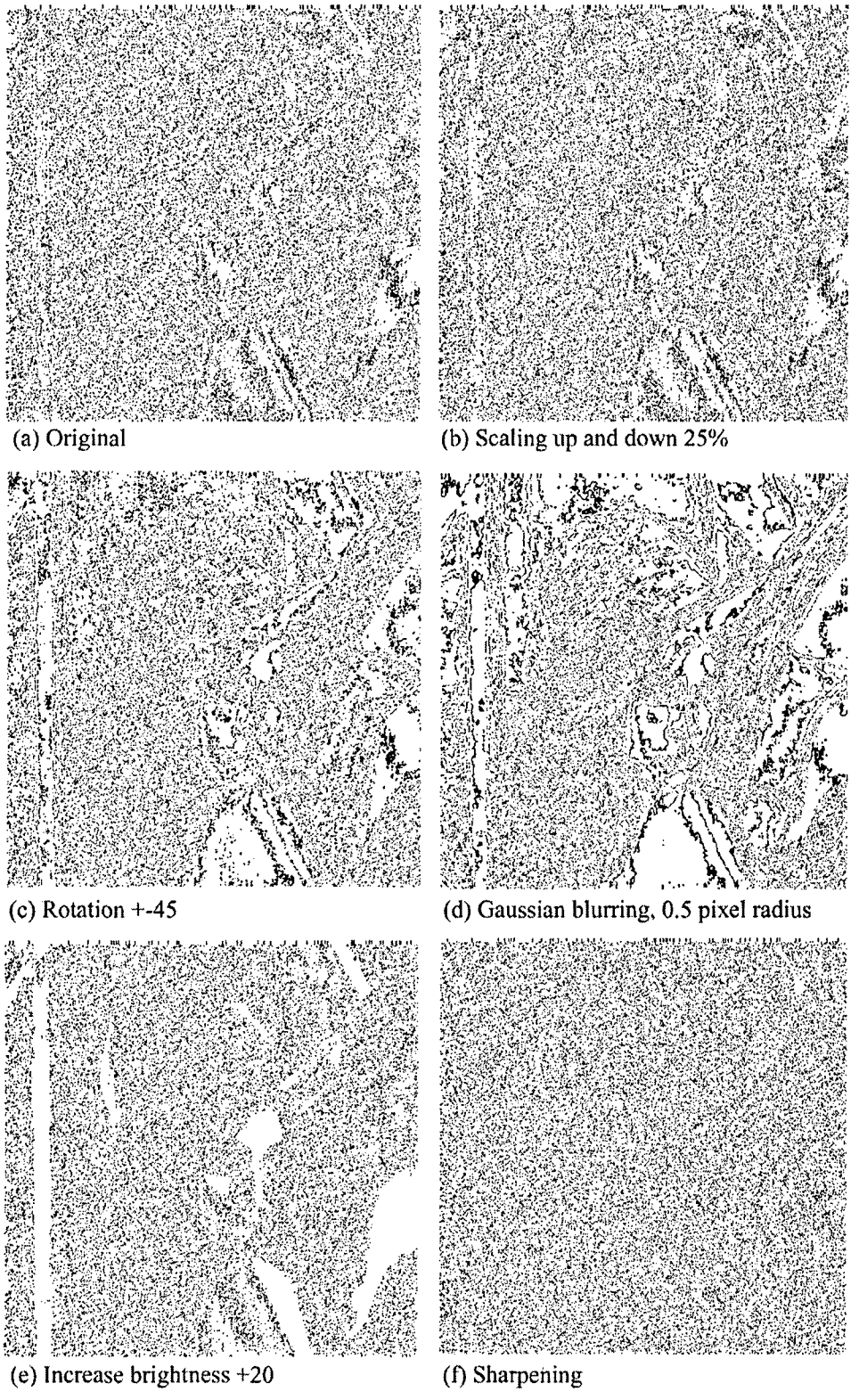


Fig. 2 Last three bit planes of the "Lena" image and its manipulated versions. The manipulation parameters in the caption are the Photoshop parameters. The range 0 to 7 in the three least significant bit planes is stretched to 0 to 255.

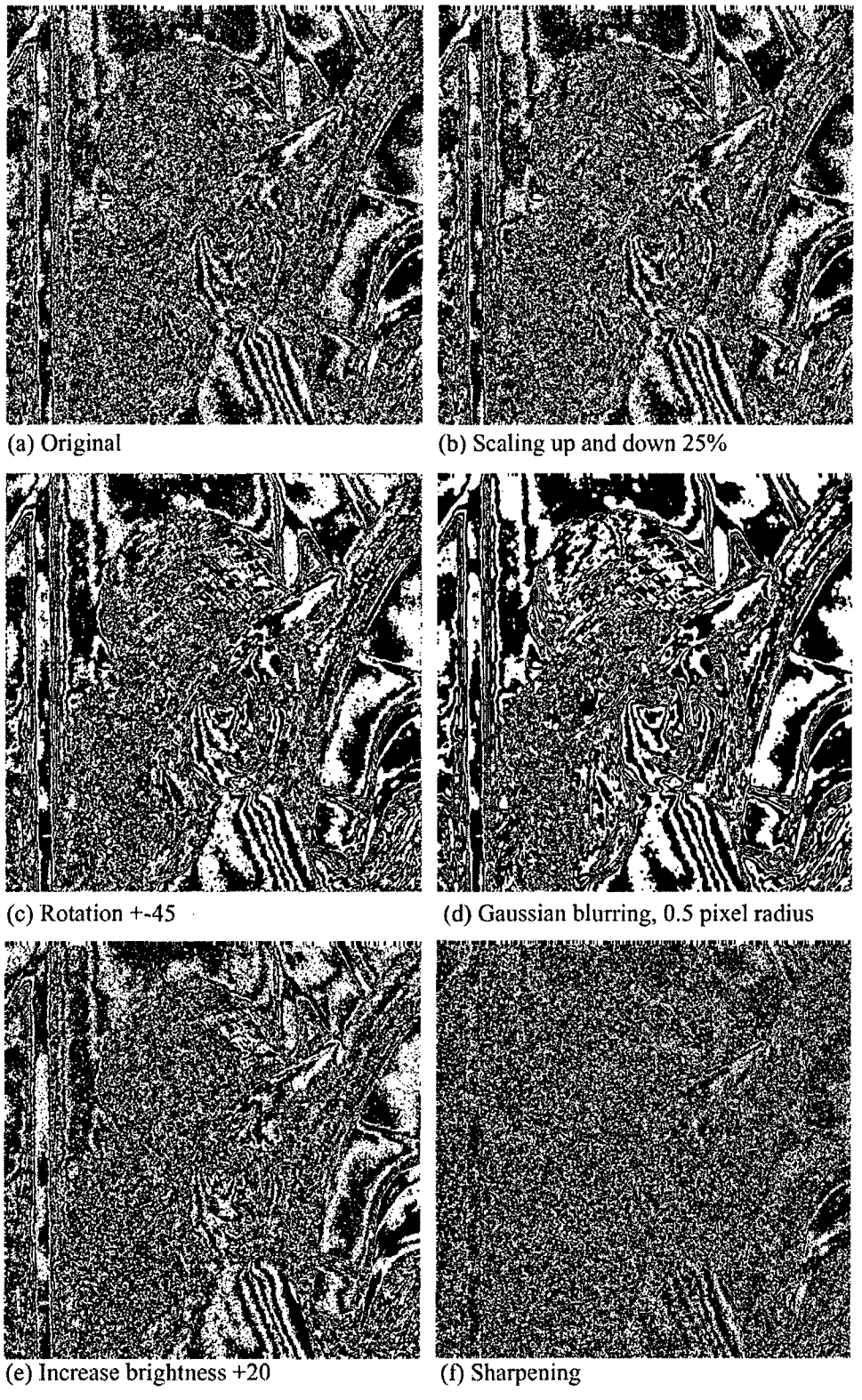


Fig. 3 Differences between the fifth and the sixth bit planes of the "Lena" image and its manipulated versions. The range $[-1,1]$ of bit plane differences is mapped to 0 to 255.

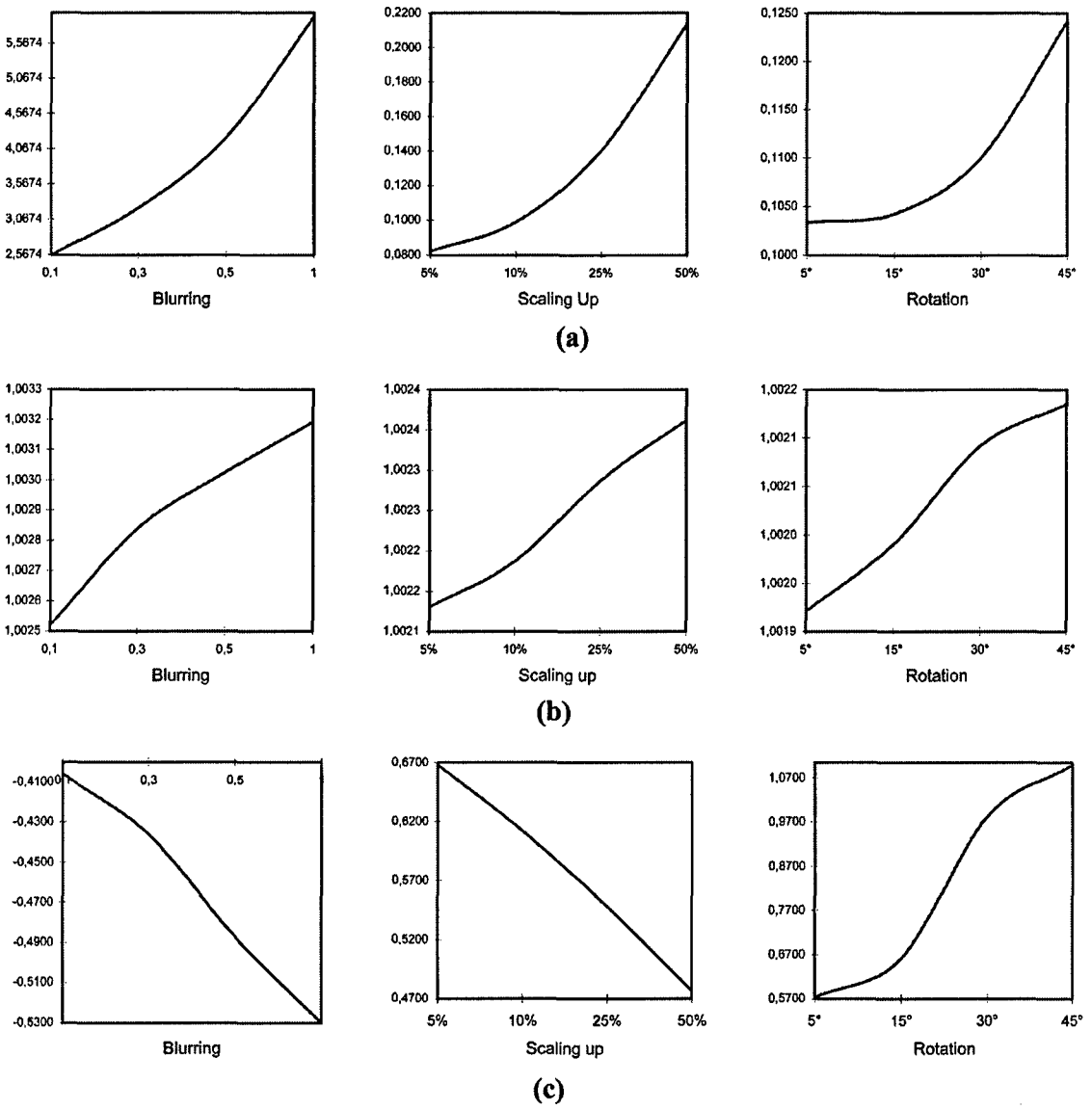


Fig. 4 (a) Sneath and Sokal measure of binary similarity versus the strength of manipulations; (b) normalized correlation measure of the IQM versus the strength of manipulations of the blurring, scaling-up, and rotation manipulations; and (c) variance of vertical subband HOWS measures versus the strength of manipulations of the blurring, scaling-up, and rotation manipulations. The Photoshop manipulation parameters are as follows: blurring, 0.1, 0.3, 0.5, and 1.0; scaling-up, 5, 10, 25, and 50%; and rotation, 5, 15, 30, and 45 deg.

The formulas for these BSMs are given in Table 1 and also reported in Ref. 12. Columns 3 to 11 indicate with a checkmark whether or not that feature was chosen by the SFFS algorithm. To give an example, the features most contributing to the "sharpen" manipulation are the Kulczynski similarity measure 1, the Sneath and Sokal similarity measures 1, 2, 3, and 5, Ochiai similarity measure, binary min histogram difference, binary absolute histogram difference, binary mutual entropy, binary Kullback-Leibler distance, Ojala mutual entropy and Ojala Kullback-Leibler distance.

In Table 1, as well as Tables 2 and 3 of IQM and HOWS features, respectively, we present in columns 3 to 9 the semi-blind cases that is, when the detector knows the specific manipulation, but not its strength. For example, the classifier is trained to differentiate between original and blurred images, while being presented with images subjected to a range of blurring degrees. The last two columns (10 and 11) of the tables require special attention. Column 10 (JFS) is the blind manipulation case, that is, the classifier does not know the type and strength of the manipulation, if there is any, and is trained with all sorts of image manipulations. Finally, column 11 (CFS) shows the features selected by the core set. Notice that the format of Table 3

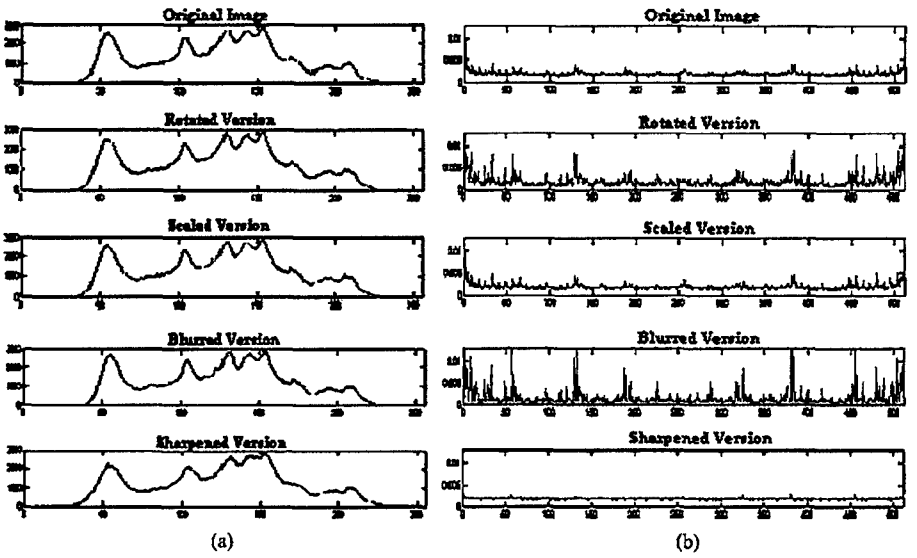


Fig. 5 Histograms of the original "Lena" image obtained in the face of image manipulations: (a) gray-level histograms and (b) Ojala histograms (normalized) from the bit planes 3 and 4.

HOWS) is different than those of Tables 1 and 2 since we are not using the SFFS procedure for the HOWS method in single semiblind sets; SFFS is used only in the CFS.

2.2 *IQMs*

IQMs were employed in Ref. 9 in the context of both passive and active warden image steganography schemes. These measures address various quality aspects of the difference image between the original and its denoised version. Among the 22 candidate measures investigated, the survivors of the SFFS selection procedure are listed in Table 2. To give a flavor of these features, the cross-correlation measure, appearing in the second row of this table, was illustrated Fig. 4(b). Notice the almost linear dependence of the cross-correlation measure versus the strength of the image manipulation operation. Different than the BSM case, where only one spectral component was used, the IQMs are calculated using all three color components. Notice, for example that, the SNR feature suffices all by itself to discriminate the "sharpen" manipulation from among the IQMs.

2.3 *HOWS*

The HOWS features[3,5,6] are obtained via a decomposition of the image using separable quadrature mirror filters. This decomposition splits the frequency domain into multiple scales and orientations, in other words, by generating vertical, horizontal, and diagonal subband components. Given the image decomposition, a first set of statistical features is obtained by the mean, variance, skewness, and kurtosis of the coefficients of the n subbands. These four moments, computed over the three orientations and n subbands make up $4\times3\times(n-1)$ features. A second set of statistics is based on the norms of the optimal linear predictor residuals. For purposes of illustration, consider first a vertical band $V_i(x,y)$ at scale I and use the notation $V_i(x,y)$, $H_i(x,y)$, and $D_i(x,y)$, respectively, for the vertical, horizontal and diagonal subbands at scale $i=1,\ldots,n$. A linear predictor for the magnitude of these coefficients can be built using the intra- and intersubband coefficients. For example, for the vertical i'th subband with w_k denoting the scalar weighting coefficients one has the residual term $e_{v,i}(x,y)$:

$$\begin{aligned}|e_{v,i}(x,y)| = {} & w_1|V_i(x-1,y)| + w_2|V_i(x+1,y)| \\ & + w_3|V_i(x,y-1)| + w_4|V_i(x,y+1)| \\ & + w_5|V_{i+1}(x/2,y/2)| + w_6|D_i(x,y)| \\ & + w_7|D_{i+1}(x/2,y/2)|.\end{aligned}$$

The prediction coefficients and hence the prediction error terms can be estimated using the Moore-Penrose inverse. An additional statistical set of $4\times3\times(n-1)$ features can be collected from the mean, vaeiance, skewness, and kurtosis of the prediction residuals at all scales. The final feature vector is $24(n-1)$ dimensional, e.g., 72 for $n=4$ scales (here $n=1$ represents the original image). When the HOWS features where subjected to the SFFS selection procedure in the blind manipulation scenario, the following 16 features were selected. Recall that the CFS and the JFS were selected from the pool of all 188 features. The 16 shown in the Table 3 represent the portions of HOWS features in the core set.

3 Experimental Results and Detection Performance

In our experiments we built a database of 200 natural images. These images were expressly taken with a single camera (Canon Powershot S200). The reason is that each camera brand possesses a different CFA, which may impact on the very features with which we want to detect alterations.[17] The database constructed with a single camera eliminates this CFA confounding factor.

The image alterations we experimented with were scaling up, scaling down, rotation, brightness adjustment, con-

Table 1 Selection of BSM features per manipulation (semiblind detector) as well as when all manipulations are presented (blind detector); for simplicity, we do not indicate the bit plan pairs used for each feature.

Similarity Measure	Description	Up	Down	Rotation	Contrast	Bright	Blurring	Sharpen	JFS	CFS
Sneath and Sokal similarity measure 1	$m_1=\frac{2(a+d)}{2(a+d)+b+c}$ $\{dm_1^{k,l}=m_1^k-m_1^l; k=3,\ldots,7, l=4,\ldots,8; \lvert k-l\rvert=1\}$ and similarly for m_1 and to m_9	√	√	√	√		√	√	√	
Sneath and Sokal similarity measure 2	$m_2=\frac{a}{a+2(b+c)}$		√	√	√		√	√		
Kulczynski similarity measure 1	$m_3=\frac{a}{b+c}$	√			√	√		√	√	
Sneath and Sokal similarity measure 3	$m_4=\frac{a+d}{b+c}$					√		√		
Sneath and Sokal similarity measure 4	$m_5=\frac{\frac{a}{(a+b)}+\frac{a}{(a+c)}+\frac{d}{(b+d)}+\frac{d}{(c+d)}}{4}$	√	√		√		√		√	√
Sneath and Sokal similarity measure 5	$m_6=\frac{ad}{[(a+b)(a+c)(b+d)(c+d)]^{1/2}}$	√		√			√	√		
Ochiai similarity measure	$m_7=\left[\left(\frac{a}{a+b}\right)\left(\frac{a}{a+c}\right)\right]^{1/2}$	√	√	√	√		√	√	√	
Binary Lance and Williams nonmetric dissimilarity measure	$m_8=\frac{b+c}{2a+b+c}$		√			√	√			
Pattern difference	$m_9=\frac{bc}{(a+b+c+d)^2}$	√	√			√	√		√	
Variance dissimilarity measure	$m_{10}=\Sigma_{n=1}^4 \min(p_n^1, p_n^2)$		√			√				
Binary min histogram difference	$dm_{11}=\Sigma_{n=1}^4 \lvert p_n^1-p_n^2\rvert$		√	√				√	√	√
Binary absolute histogram difference	$dm_{12}=-\Sigma_{n=1}^4 p_n^1 \log p_n^2$	√	√	√	√			√	√	√
Binary mutual entropy	$dm_{13}=-\Sigma_{n=1}^4 p_n^1 \log \frac{p_n^1}{p_n^2}$		√				√	√	√	
Binary Kullback-Leibler distance	$dm_{14}=\Sigma_{n=1}^N \min(S_n^1, S_n^2)$		√		√	√	√	√	√	√
Ojala min histogram difference	$dm_{15}=\Sigma_{n=1}^N \lvert S_n^1-S_n^2\rvert$		√		√	√	√		√	
Ojala absolute histogram difference	$dm_{16}=-\Sigma_{n=1}^N S_n^1 \log S_n^2$		√	√			√		√	
Ojala mutual entropy	$dm_{17}=-\Sigma_{n=1}^N S_n^1 \log \frac{S_n^1}{S_n^2}$		√	√	√			√	√	
Ojala Kullback-Leibler distance	$dm_{18}=-\Sigma_{n=1}^N S_n^7 \log \frac{S_n^7}{S_n^8}$	√	√	√		√	√	√	√	√

Table 2 Selected IQM features; here C_k and $\hat{C}_k$ denote, respectively, the original and blurred versions of the k'th spectral component, $k=1,\ldots,K$ of some image; other symbols are defined in Appendix B.

Similarity Measure	Description	Up	Down	Rotation	Contrast	Bright	Blurring	Sharpen	JFS	CFS
Mean square error (MSE)	$D1=\frac{1}{K}\sum_{k=1}^{K}\left[\frac{1}{N^2}\sum_{i,j=1}^{N}\left\|Ck(i,j)-\hat{C}k(i,j)\right\|^2\right]^{1/2}$	√		√	√	√	√	√	√	√
Cross-correlation measure	$D2=\frac{1}{K}\sum_{k=1}^{K}\frac{\sum_{i,j=0}^{N-1}Ck(i,j)\hat{C}k(i,j)}{\sum_{i,j=0}^{N-1}Ck(i,j)^2}$			√	√		√	√		
Laplacian MSE	$D3=\frac{1}{K}\sum_{k=1}^{K}\left[\frac{1}{N^2}\sum_{i,j=1}^{N}\left\|Ck(i,j)-C'k(i,j)\right\|^2\right]^{1/2}$	√				√	√	√		√
Mean angle similarity	$D4=\mu_x=1-\frac{1}{N^2}\sum_{i,j=1}^{N}\left[\frac{2}{\pi}\cos^{-1}\frac{\langle C(i,j),\hat{C}(i,j)\rangle}{\|C(i,j)\|\|\hat{C}(i,j)\|}\right]$		√	√		√	√	√	√	
Mean angle-magnitude similarity	$D5=\frac{1}{N^2}\sum_{i,j=1}^{N}\chi ij$			√	√	√	√	√	√	√
(HVS) L2 Norm	$D6=\frac{1}{K}\sum_{k=1}^{K}\frac{\sum_{i,j=0}^{N-1}\|U[Ck(i,j)]-U[\hat{C}k(i,j)]\|}{\sum_{i,j=0}^{N-1}\|U[Ck(i,j)]\|}$		√	√	√	√	√			√
Spectral phase error	$D7=\frac{1}{N^2}\sum_{u,v=0}^{N-1}\|M(u,v)-\hat{M}(u,v)\|^2$	√	√	√			√	√		
Spectral phase-magnitude error	$D8=\frac{1}{N^2}\sum_{u,v=0}^{N-1}\|\varphi(u,v)-\hat{\varphi}(u,v)\|^2$		√			√			√	

trast enhancement, blurring, and sharpening, all implemented via Adobe Photoshop.[18] For example, in the scaling-up manipulation, images were enlarged by six factors, namely, by 1, 2, 5, 10, 25, and 50%, resulting in 1200 scaled-up images. In all cases, half of the images were randomly selected for training, and the remaining half was used for testing. Thus, the total image database with manipulations climbed up to 6200. Table 4 lists the manipulations and their strength parameters.

3.1 *Assessment of Feature Sets*

We ran experiments to assess the power of feature sets in all modes, namely, clairvoyant, semiblind, and blind.

3.1.1 *Clairvoyant mode*

In this case, the detector is aware of the specific manipulation as well as of its strength. Figure 6 illustrates the relative competition between each method (clairvoyant mode)

Table 3 Selected HOWS features in the core set.

	Vertical Subband			Horizontal Subband			Diagonal Subband		
Scale	1	2	3	1	2	3	1	2	3
Mean		√	√	√	√			√	
Variance	√						√	√	
Kurtosis		√							√
Skewness									
Mean of linear prediction error					√				
Variance of linear prediction error				√			√		√
Kurtosis of linear prediction error							√		
Skewness of linear prediction error							√		

Table 4 Selected Photoshop image manipulations and their parameters.

Scaling up (%)	1	2	5	10	25	50
Scaling down (%)	1	5	10	25	50	
Rotation (deg)	1	5	15	30	45	
Contrast enhancement	1	5	10	25		
Brightness adjustment	5	15	25	40		
Blurring (with radius)	0.1	0.3	0.5	1.0	2.0	
Sharpen			Photoshop Default			

against different manipulation types, and the JFS that outperforms all. One can see from this figure that, while the performance of methods may vary much from manipulation to manipulation, the JFS is always the best performing one. More explicitly, the SFFS was run separately for the IQM, BSM, HOWS, and JFS sets, and each feature set was optimized for the specific manipulations. Figure 7 illustrates the competition between feature categories, where a separate detector was trained for each manipulation type, but with unknown strength. Here we limit ourselves to two illustrative cases, one where HOWS outperform the others (manipulation by rotation) and another where BSM outperforms all others (manipulation by contrast enhancement).

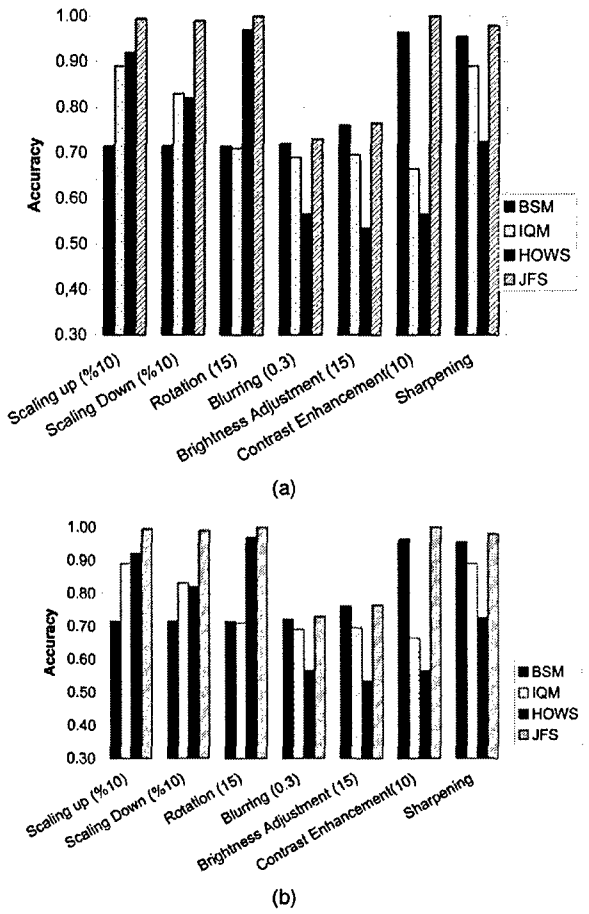


Fig. 6 Comparative performance of feature sets optimized for various manipulations. For comparison purposes, one midlevel manipulation strength was chosen, as denoted next to each manipulation.

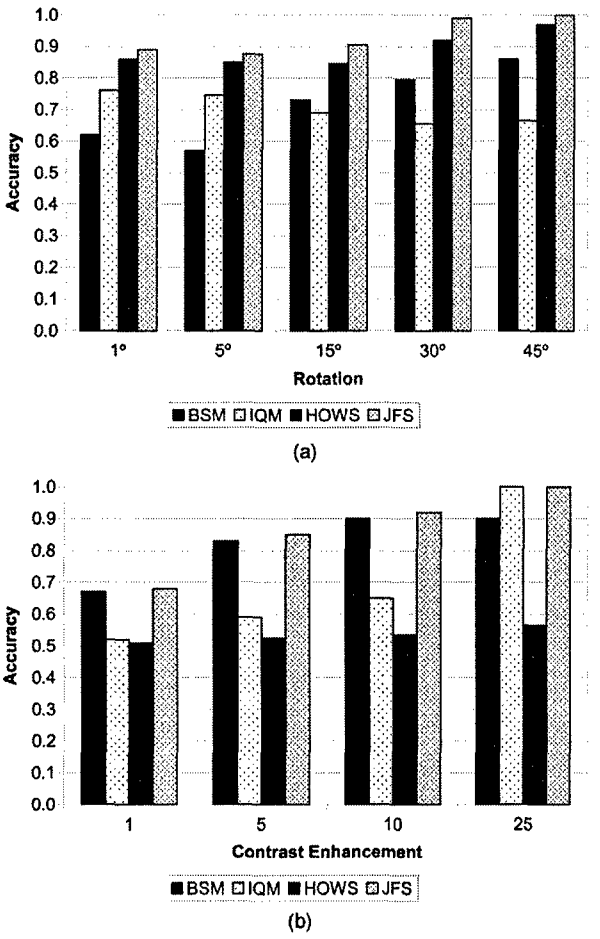


Fig. 7 Performance of clairvoyant classifiers for (a) rotation manipulation and (b) contrast enhancement manipulation.

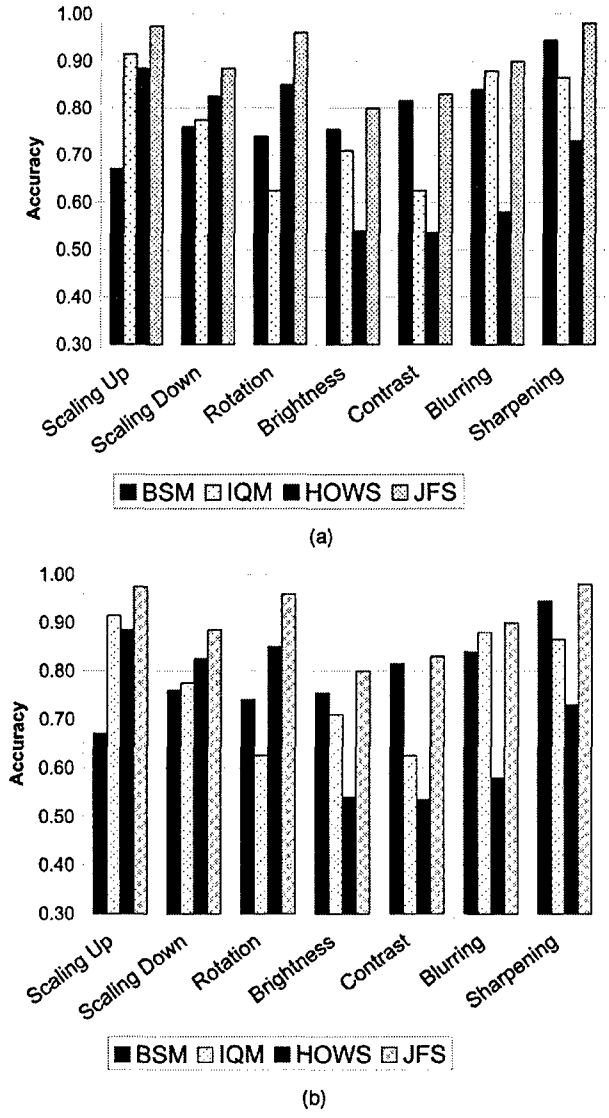


Fig. 8 Performance of semi-blind classifiers, which are independent of the strength of manipulation.

Notice that the fourth bar has the richest feature selection from BSM+IQM+HOWS, hence it is always better.

3.1.2 *Semiblind mode*

In this case, the detector is aware of the specific manipulation, but not of its strength, which could vary as in Table 4. For example, we generate a separate image pool from the settings of 25, 10, 5, and 2% scaling-up parameter, and train the corresponding "scaling-up forensic detector." The SFFS outcomes for the BSM and IQM sets were already exemplified in Tables 1–3, respectively. One can notice in Fig. 8 that each feature category has its own strengths and weaknesses vis-à-vis the manipulations and that it is not possible to determine a single winner category for all cases. However, when we allow the selection algorithm to pick up features from the pooled IQM, BSM, and HOWS sets, the detection performances are always better. It is intriguing to discover the proportions of feature categories taking role in the classifications. The percentages of features derived from respective BSM, IQM, and HOWS categories are illustrated in Fig. 9. The enrollment percentages of categories vary widely from case to case, though in general the BSM and HOWS features dominate

3.1.3 *Blind mode*

Finally we designed blind classifiers, that is, classifiers that are independent of the manipulation type and of its strengths. This classifier was designed using training and test sets incorporating images that were manipulated with all items in Table 4. The corresponding performance results of this blind classifier are given in Figs. 10 and 11. The following comments are in order:

1. The classifier designed from pooled features outperforms the classifiers that were designed with features from a single category, be it BSM, IQM, or HOWS (Fig. 10). As expected, the JFS, which optimizes its feature set and classifier weight for the occasion, is better than the CFS, which can only optimize its classifier weights for a fixed set of features.
2. The left bars (JFS) in Fig. 11 denote the performance of semiblind classifiers (specific for a manipulation). Note here that there is not much of a performance difference between the JFS trained to detect all manipulations, as in Fig. 10, and the JFSs trained to detect just one specific manipulation. Obviously these bars are the same as the rightmost bars in Fig. 8.
3. The right bars (CFS), on the other hand, correspond to the classifier performance when the core subset of features were used, but trained separately for each type of manipulation. In this case, only the weights differ in the regression equations, but the type and number of features are the same. The results with the CFS are slightly inferior to the JFS set, as expected. The bar in Fig. 11 denotes the average of the CFS performance.
4. Figure 12 shows pie charts of features derived from the three categories. The left figure shows the portions of the three categories in the JFS case and the right in the CFS case. Notice that the BSM features dominate in the CFS, while in the JFS case the HOWS features dominate.

Finally, it would have been desirable to establish patterns or trends in the selection of features against specific manipulations for a better intuitive understanding of the problem. However, no clear trend was observable.

3.2 *Performance with Uncontrolled Experiments*

To test our scheme in a more realistic environment, we considered images doctored by extra content insertion and replacement of the original content, e.g., as in Fig. 1. This is in contrast to the experiments in Sec. 3.1, where the whole image was subjected to one manipulation type at a time. Instead, in uncontrolled experiments, sequences of image manipulation take place in image patches, with possible forgery intention.

In the first round of experiments, we used a set of 20 images, all captured with the same camera to preclude the

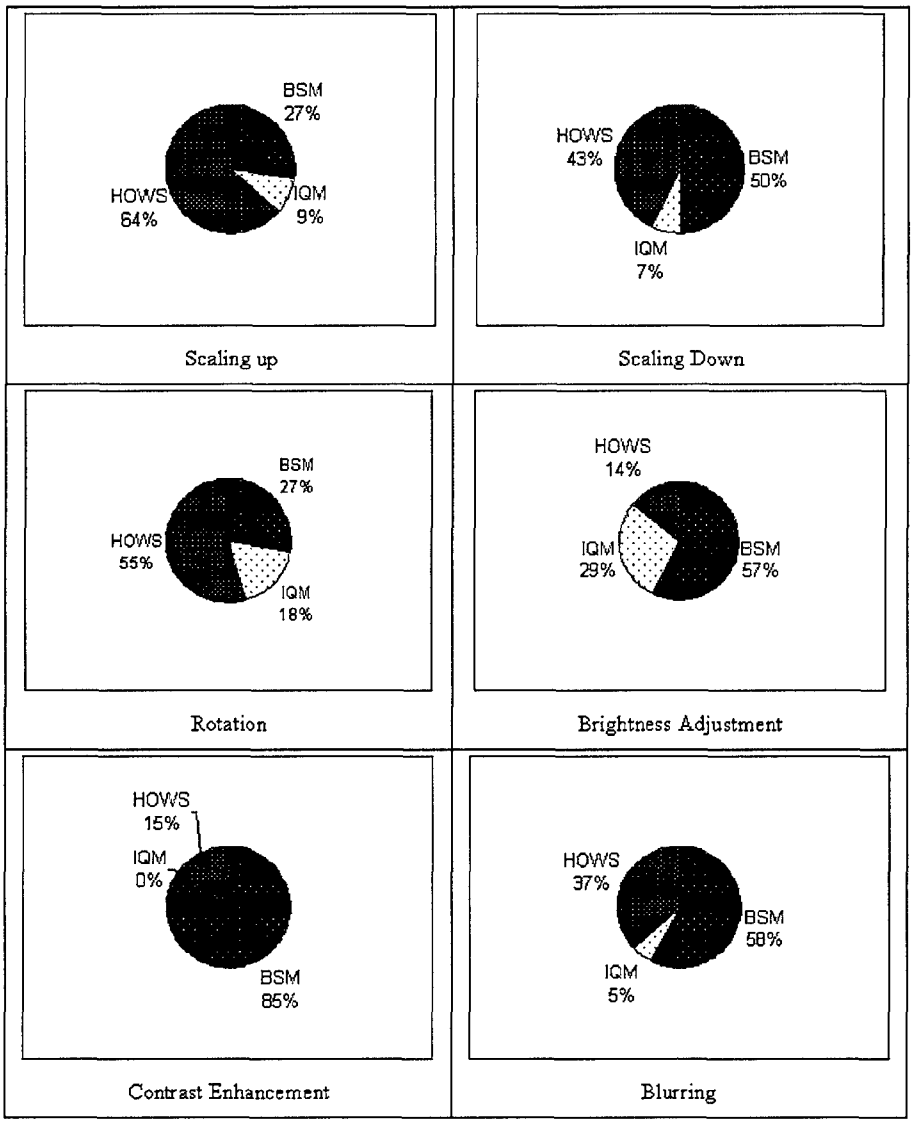


Fig. 9 Enrollment percentages from the three feature categories.

nuisance of different camera parameters. We spent some effort to make them look like natural images to avoid any suspicion. To this effect, the inserted content was resized, rotated and/or brightness adjusted before being pasted onto the image, the parameters of these manipulations being adjusted with visual fitness criteria. Sample images that have undergone doctoring operations are shown in Fig. 13. We took two untampered and one tampered block from every image, to create a repertoire of 40 untampered and 20 tampered blocks. The block sizes were varying but the smallest block size was 100×100, while the original image sizes were 640×480. The manipulated regions of the image, like the heads in Figs. 1 and 13(a) and the people in Fig. 13(b), fitted into the block size. Notice that one does not exhaustively search with blocks positioned over all possible pixels. Instead, one is tempted to test the suspected regions, like persons, faces, etc.

We tested these images using all the six semiblind, that is, manipulation-specific, classifiers, since any one or more of the manipulation types could have taken place. We declared "a manipulation has occurred" in a block whenever any one of the semiblind detectors gave an alarm. In other words, the binary decision was taken with decision fusion from the six manipulation experts using the binary sum rule. False positives occur if an untampered block is erroneously declared as "manipulated"; similarly, false negatives result when all six experts fail to see evidence of manipulation for a block that was actually manipulated. Table 5 shows the results for the image blocks on generic classifiers. The performance of blind classifiers are listed in Table 6.

As a further proof of the viability of our scheme, we captured 100 images from the Internet with obvious tampering clues. Sample images are displayed in Fig. 14. We tested these images on semi-blind and blind classifiers (see

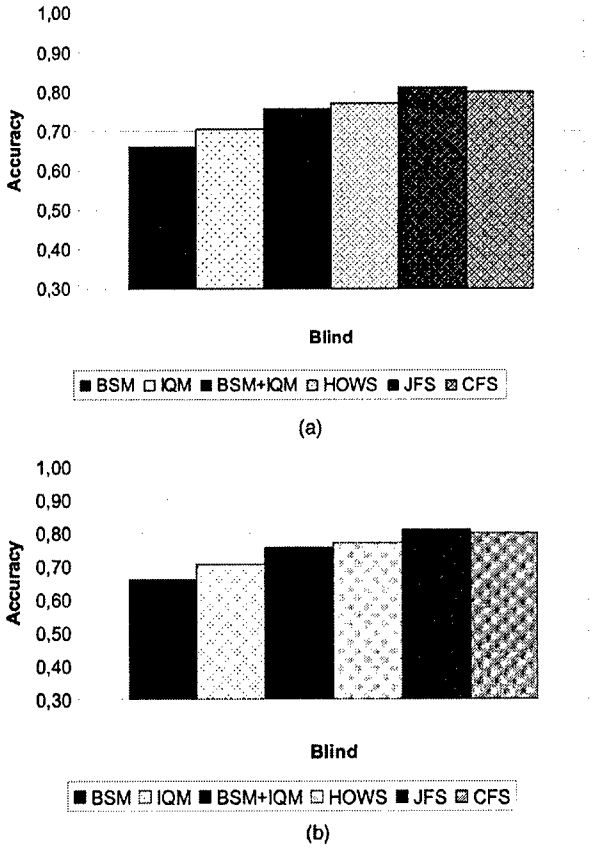


Fig. 10 Performance of blind classifiers. Each bar denotes a different way of selecting features. The first three bars denote the feature set when we limit the choice to the respective BSM, IQM, and HOWS categories. Both the JFS and the CFS select from all categories but in two different styles, as explained in Sec. 3.

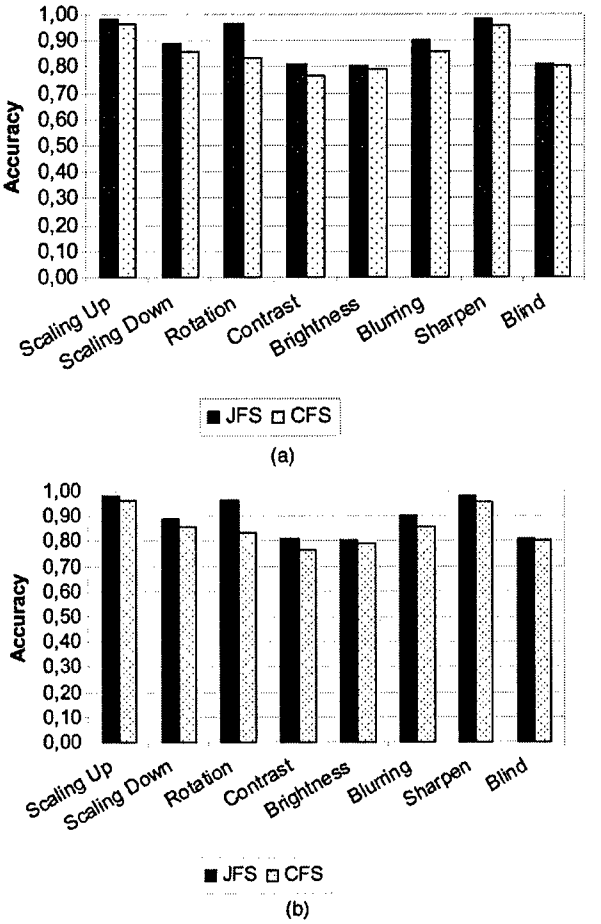


Fig. 11 Performance of semiblind and blind classifiers. The left bars denote the strength-blind manipulation-clairvoyant classifier, where both features and regression coefficients could be trained; finally, the right bars denote the strength-blind, manipulation-clairvoyant classifier, where the core of features were common and fixed but regression coefficients could be trained.

Tables 7 and 8). Notice that for the images downloaded from the Internet, the tests are on the whole image, and not on a block basis.

4 Conclusions

We developed an image forensic scheme based on the interplay between feature fusion and decision fusion. We considered three categories of features, namely, the binary similarity measures between the bit planes, the image quality metrics applied to denoised image residuals, and the statistical features obtained from the wavelet decomposition of an image. These forensic features were tested against the background of single manipulations and multiple manipulations, as would actually occur in doctoring images. In the first set of single-manipulation experiments, we observed that each feature category has its weak and strong points vis-à-vis manipulation types, and that it is best to select features from the general pool of all categories (feature fusion). In the second set of experiments with multiple manipulations, the best strategy was to use different types of classifiers (experts) one per manipulation, and then fuse their decisions.

Further issues that remain to be explored are as follows: (1) The implementation of the decision fusion with alternative schemes, such as max rule, sum rule, ranked voting, or weighted plurality voting;[19] (2) investigation of a more general set of manipulation tools that are instrumental in image doctoring; and (3) singular value decomposition and nonnegative matrix factorization are powerful tools for matrix analysis, and its potential directly reflects on images, when image blocks are viewed as nonnegative matrices.

One intriguing question is whether and how to create images and image manipulations that will go through undetected by our scheme, especially if all the measures of

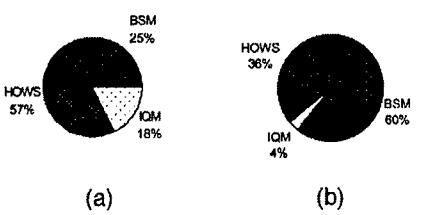


Fig. 12 Pie charts of feature sets for the JFS (left) and CFS (right) in blind mode.

Fig. 13 Examples of doctored images: (a) Changing the content of the image (left genuine, right forged) and (b) adding extra content to an image (left genuine, right forged).

Fig. 14 Examples of the images captured from the Internet. The top left of the image is the original one. All other images are doctored.

the method are publicly available. Figures 7–10 give some clues as to the probability of avoiding being caught. Some manipulations are more easily detected; for example, Fig. 7(a) shows that the rotation expert is able detect even 1 deg of rotation with 88% success in clairvoyant mode; on the other hand, the success rate for contrast enhancement experts is inferior. As can be expected from any doctoring detector, our approach also has a weak belly to very small levels of manipulations. On the other hand, only objective psychovisual measures can decide at what point the doctoring effects impact on the semantic content.

5 Appendix A: BSM Features

Let $\mathbf{x}_i=\{x_{i-k}|,k=1,\ldots,K\}$ and $\mathbf{y}_i=\{y_{i-k}|,k=1,\ldots,K\}$ be the sequences of bits representing the K-neighborhood pixels, where the index i runs over all the $M\times$N image pixels. For $K=4$, we obtain the four stencil neighbors over which we define the indicator function as

$$\chi_{r,s}=\begin{cases}1 & \text{if } x_r=0 \quad \text{and } x_s=0\\ 2 & \text{if } x_r=0 \quad \text{and } x_s=1\\ 3 & \text{if } x_r=1 \quad \text{and } x_s=0\\ 4 & \text{if } x_r=1 \quad \text{and } x_s=1.\end{cases}$$

Thus, the agreement variable for the pixel x_i is obtained as $\alpha_i^j=\sum_{k=1}^{K}\delta(\chi_{i,i-k},j)$, $j=1,\ldots,4$, $K=4$, where δ is the Dirac delta selector. Finally, the accumulated agreements can be defined as

$$a=\frac{1}{MN}\sum_i \alpha_i^1, \quad b=\frac{1}{MN}\sum_i \alpha_i^2,$$

Table 5 Performance of semiblind classifiers for image blocks.

Method	False Positive	False Negative	Accuracy (%)
BSM	20/40	2/20	63.33
IQM	25/40	2/20	55.00
BSM+IQM	9/40	2/20	81.67
HOWS	40/40	0/20	33.33
CFS	6/40	3/20	85.00
JFS	**5/40**	**0/20**	**91.67**

Table 6 The performance of blind classifiers for image blocks.

Method	False Positive	False Negative	Accuracy (%)
BSM	19/40	6/20	58.33
IQM	23/40	4/20	55.00
BSM+IQM	8/40	4/20	80.00
HOWS	9/40	8/20	71.67
JFS	**1/40**	**5/20**	**90.00**

Table 7 The performance of semiblind classifiers for image blocks that are captured from the Internet.

Method	False Negative	Accuracy (%)
BSM	21/100	79
IQM	19/100	81
BSM+IQM	9/100	91
HOWS	0/100	100
JFS	**0/100**	**100**

$$c=\frac{1}{MN}\sum_i \alpha_i^3, \quad d=\frac{1}{MN}\sum_i \alpha_i^4.$$

These four variables $\{a,b,c,d\}$ can be interpreted as the one-step cooccurrence values of the binary images. Obviously these cooccurrences are defined for a specific bit plane b, though the bit plane parameter was not shown for the sake simplicity. Normalizing the histograms of the agreement scores for the bth bit plane [where now $\alpha_i^j = \alpha_i^j(b)$] one obtains for the j'th cooccurrence:

$$p_j^\beta=\frac{\sum_i \alpha_i^j}{\sum_i \sum_j \alpha_i^j}; \quad \beta=3,\ldots,8.$$

Three categories of similarity measures are derived from the local bit plane features, as detailed next.

The first group of features uses various functional combinations of local binary texture measures. Actually, as pointed out in the first row of the Table 1, the differential measure $dm_i^{k,l}=m_i^k-m_i^l$ over adjacent bit plane pairs, the kth and the lth, is used. The feature selection algorithm selects none, one, or more of the appropriate bit plane pairs $\{dm_i^{k,l}=m_i^k-m_i^l; \quad k=3,\ldots,7, \quad l=4,\ldots,8; \quad |k-l|=1, i=1,\ldots,10\}$ that are found to be the most effective in classification. In Table 1, therefore, we do not indicate the specific bit planes used, since these are to be chosen adaptively by the feature selection algorithm. Thus, this first group results in 60 features, since there are 10 varieties, each computed over six adjacent bit plane pairs. The chosen bit plane pairs vary from manipulation to manipulation; for example, blurring demands dm_2 between bit planes 7-8 and 3-4, that is, $\{dm_2^{3,4}, dm_2^{7,8}\}$.

Table 8 The performance of blind classifiers for image blocks that are captured from the Internet.

Method	False Negative	Accuracy (%)
BSM	58/100	42
IQM	51/100	49
BSM+IQM	48/100	52
HOWS	47/100	53
JFS	**11/100**	**89**

1	2	4
128	x_i (256)	8
64	32	16

(a)

0	1	0
1	0	0
0	1	1

(b)

Fig. 15 (a) Weighting of the neighbors in the computation of Ojala score and (b) example of Ojala score for the given bit pattern where the central bit is 0: $S=2+16+32+128=178$.

A second group of features consists of histogram and entropic features. Based on normalized four-bin histograms, we define the minimum histogram difference dm_{11} and the absolute histogram difference measures dm_{12}, binary mutual entropy dm_{13}, and binary Kullback-Leibler distance dm_{14}, as also given in Table 1. There are therefore overall 24 such features defined over the six bit plane pairs.

The third set of measures, $dm_{14},\ldots,dm_{17}$ are somewhat different in that we use the neighborhood-weighting mask proposed by Ojala. For each binary image we obtain a 512-bin histogram using directional weighting of the eight neighbors. We have in total 24 features, with four varieties computed over six bit planes. To give a flavor of, binary similarity measures we consider the Ojala[18] histograms. For each binary image on the bth bit plane we obtain a 512-bin histogram based on the weighted eight neighborhood, as in Fig. 15. For each eight-neighborhood pattern, the histogram bin numbered is augmented by 1.

Finally, the entropic measures are defined as follows. Let the two normalized histograms be denoted as S_n^β, $n=0,\ldots,255$ and $\beta=3,\ldots,7$. The resulting Ojala measure is the mutual entropy between the two distributions belonging to adjacent planes b and $b+1$:

$$m_\beta=-\sum_{n=1}^{N} S_n^\beta \log S_n^{\beta+1}.$$

6 Appendix B: IQM Features

In this appendix we define and describe image quality measures considered. In these definitions the pixel lattices of images A and B are referred to as $A(i,j)$ and $B(i,j)$, $i,j=1,\ldots,N$, as the lattices are assumed to have dimensions $N\times N$. The pixels can take values from the set $\{0,\ldots,255\}$. Similarly, we denote the multispectral components of an image at the pixel position i,j, and in band k, as $C_k(i,j)$, where $k=1,\ldots,K$. The boldface symbols $\mathbf{C}(i,j)$ and $\hat{\mathbf{C}}(i,j)$ indicate the multispectral pixel vectors at position (i,j). For example, for the color images in the RGB representation one has $\mathbf{C}(i,j)=[R(i,j)G(i,j)B(i,j)]^T$. All these definitions are summarized in Table 9.

Thus, for example, the power in the k'th band can be calculated as $\sigma_k^2=\Sigma_{i,j=0}^{N-1}C_k(i,j)^2$. All these quantities with an additional hat, i.e., $\hat{C}_k(i,j)$, $\hat{\mathbf{C}}$ etc., correspond to the dis-

Table 9 Summary of definitions for IQM features.

Symbol	Definition
$C_k(i,j)$	(i,j) th pixel of the k th band of image C
$\mathbf{C}(i,j)$	(i,j) th multispectral (with K bands) pixel vector
$\mathbf{C}$	multispectral image
C_k	k th band of multispectral image C
$\varepsilon_k = C_k - \hat{C}_k$	error over all the pixels in the k th band of multispectral image C

torted versions of the same original image. As a case in point, the expression $\|\mathbf{C}(i,j)-\hat{\mathbf{C}}(i,j)\|^2=\Sigma_{k=1}^{K}[C_k(i,j)-\hat{C}_k(i,j)]^2$ denotes the sum of errors in the spectral components at a given pixel position i,j. Similarly, the error expression in the last row of Table 9 expands as $\varepsilon_k^2=\Sigma_{i=1}^{N}\Sigma_{j=1}^{N}[C_k(i,j)-\hat{C}_k(i,j)]^2$. In the specific case of *RGB* color images, we occasionally revert to the notations $\{R,G,B\}$ and $\{\hat{R},\hat{G},\hat{B}\}$.

Quality metrics can be categorized into six groups according to the type of information they use.[10] The categories used are

1. pixel-difference-based measures such as mean square distortion
2. correlation-based measures, that is, correlation of pixels, or of the vector angular directions
3. edge-based measures, that is, displacement of edge positions or their consistency across resolution levels
4. spectral distance-based measures, that is, Fourier magnitude and/or phase spectral discrepancy on a block basis
5. context-based measures, that is penalties based on various functionals of the multidimensional context probability
6. HVS-based measures, measures either based on the HVS-weighted spectral distortion measures or (dis-)similarity criteria used in image base browsing functions.

6.1 *Pixel-Difference-Based Measures*

6.1.1 *Mean square error*

$$D_1=\frac{1}{K}\sum_{k=1}^{K}\left[\frac{1}{N^2}\sum_{i,j=1}^{N}|Ck(i,j)-\hat{C}k(i,j)|^2\right]^{1/2},$$

where $K=3$ for *RGB* color images.

6.2 *Correlation Based Measures*

6.2.1 *Normalized cross-correlation measure*

The closeness between two digital images can be quantified in terms of the normalized cross-correlation function:

$$D_2=\frac{1}{K}\sum_{k=1}^{K}\frac{\sum_{i,j=0}^{N-1}C_k(i,j)\hat{C}_k(i,j)}{\sum_{i,j=0}^{N-1}C_k(i,j)^2}.$$

6.2.2 *Mean angle similarity*

A variant of correlation-based measures can be obtained by considering the statistics of the angles between the pixel vectors of the original and distorted images. Similar "colors" will result in vectors pointing in the same direction, while significantly different colors will point in different directions in the $\mathbf{C}$ space. Since we deal with positive vectors $\mathbf{C}$ and $\hat{\mathbf{C}}$, we are constrained to one quadrant of the Cartesian space. Thus, the normalization factor of $2/\pi$ is related to the fact that the maximum difference attained will be $\pi/2$. The combined angular correlation and magnitude difference between two vectors can be defined as follows:[20]

$$\chi_{ij}=1-\left[1-\frac{2}{\pi}\cos^{-1}\frac{\langle\mathbf{C}(i,j),\hat{\mathbf{C}}(i,j)\rangle}{\|\mathbf{C}(i,j)\|\|\hat{\mathbf{C}}(i,j)\|}\right]\times\left[1-\frac{\|\mathbf{C}(i,j)-\hat{\mathbf{C}}(i,j)\|}{\sqrt{3\cdot 255^2}}\right].$$

We can use the moments of the spectral (chromatic) vector differences as distortion measures. To this effect we have used the mean of the angle difference (D_4) and the mean of the combined angle-magnitude difference (D_5) as in the following two measures:

$$D_4=\mu_x=1-\frac{1}{N^2}\sum_{i,j=1}^{N}\left(\frac{2}{\pi}\cos^{-1}\frac{\langle C(i,j),\hat{C}(i,j)\rangle}{\|C(i,j)\|\|\hat{C}(i,j)\|}\right),$$

$$D_5=\frac{1}{N^2}\sum_{i,j=1}^{N}\chi_{ij}.$$

6.3 *Edge-Based Measures*

The edges form the most informative part in images. Some examples of edge degradations are discontinuities in the edge, decrease of edge sharpness by smoothing effects, offset of edge position, missing edge points, falsely detected edge points, etc.

6.3.1 *Laplacian mean square error:*[21]

$$D_3=\frac{1}{K}\sum_{k=1}^{K}\frac{\sum_{i,j=0}^{N-1}\{O[C_k(i,j)]-O[\hat{C}_k(i,j)]\}^2}{\sum_{i,j=0}^{N-1}\{O[C_k(i,j)]\}^2},$$

where $O[C_k(i,j)]=C_k(i+1,j)+C_k(i-1,j)+C_k(i,j+1)+C_k(i,j-1)-4C_k(i,j)$.

6.4 HVS-Based Measure

To obtain a closer relation with the assessment by the HVS, both the original and distorted images can be preprocessed via filters that simulate the HVS. One of the models for the HVS is given as a bandpass filter with a transfer function in polar coordinates:[22]

$$H(\rho)=\begin{cases}0.05\exp(\rho^{0.554}). & \rho<7\\ \exp[-9(|\log_{10}\rho-\log_{10}9|)^{2.3}] & \rho\geq 7,\end{cases}$$

where $\rho=(u^2+v^2)^{1/2}$. An image processed through such a spectral mask and then inverse discrete cosine transform (DCT) transformed can be expressed via the $U[\cdot]$ operator, i.e.,

$$U[C(i,j)]=\mathrm{DCT}^{-1}\{H[(u^2+v^2)^{1/2}]\Omega(u,v)\},$$

where $\Omega(u,v)$ denotes the 2-D DCT of the image and DCT^{-1} is the 2-D inverse DCT.

6.4.1 Normalized absolute error

$$D_6=\frac{1}{K}\sum_{k=1}^{K}\frac{\sum_{i,j=0}^{N-1}|U[C_k(i,j)]-U[\hat{C}_k(i,j)]|}{\sum_{i,j=0}^{N-1}|U[C_k(i,j)]|}.$$

6.5 Spectral Distance Measures

In this category, we consider the distortion penalty functions obtained from the complex Fourier spectrum of images. Let the DFTs of the k'th band of the original and coded image be denoted by $\Gamma_k(u,v)$ and $\hat{\Gamma}_k(u,v)$, respectively. The spectra are defined as

$$\Gamma_k(u,v)=\sum_{m,n=0}^{N-1}C_k(m,n)\exp\left(-2\pi im\frac{u}{N}\right)\times\exp\left(-2\pi in\frac{v}{N}\right),\quad k=1,\ldots,K.$$

Let phase and magnitude spectra is defined as

$$\varphi(u,v)=\arctan[\Gamma(u,v)],$$

$$M(u,v)=|\Gamma(u,v)|,$$

repectively.

6.5.1 Spectral magnitude distortion

$$D_7=\frac{1}{N^2}\sum_{u,v=0}^{N-1}|M(u,v)-\hat{M}(u,v)|^2.$$

6.5.2 Spectral phase distortion

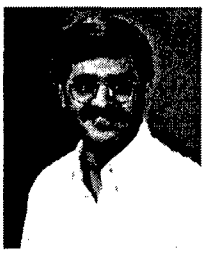
$$D_8=\frac{1}{N^2}\sum_{u,v=0}^{N-1}|\varphi(u,v)-\hat{\varphi}(u,v)|^2.$$

Acknowledgments

This work has been supported by TUBITAK under research grant KARIYER 104E056. One of the authors (N.M.) was partially supported by a grant from AFOSR

References

1. *IEEE Trans. Signal Process.* **41**(6), special issue on data hiding (2003).
2. P. Blythe and J. Fridrich, "Secure digital camera," in *Proc. Digital Forensic Research Workshop* (2004).
3. A. C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," *IEEE Trans. Signal Process.* **53**(2), 758–767 (2005).
4. I. Avcıbaş, S. Bayram, N. Memon, M. Ramkumar, and B. Sankur, "A classifier design for detecting image manipulations," in *Proc. IEEE Int. Conf. on Image Processing*, Singapore, Vol. 4, pp. 2645–2648 (2004).
5. M. K. Johnson and H. Farid, "Exposing digital forgeries by detecting inconsistencies in lighting," in *Proc. ACM Multimedia and Security Workshop*, New York, pp. 1–10 (2005).
6. A. C. Popescu and H. Farid, "Exposing digital forgeries in color filter array interpolated images," *IEEE Trans. Signal Process.* **53**(10), 3948–3959 (2005).
7. J. Fridrich, D. Soukal, and J. Lukas, "Detection of copy-move forgery in digital images," in *Proc. of the Digital Forensic Research Workshop*, Cleveland, OH (2003).
8. I. Avcıbaş, N. Memon, and B. Sankur, "Image stegalysis with binary similarity measures," *Proc. of Int. Conf. on Image Processing*, Vol. 3, pp. 645–648, Rochester (2002).
9. I. Avcibas, B. Sankur, and N. Memon, "Steganalysis of watermarking and steganographic techniques using image quality metrics," *IEEE Trans. Image Process.* **12**(2), 221–229 (2003).
10. I. Avcibas, B. Sankur, and K. Sayood, "Statistical evaluation of image quality measures," *J. Electron. Imaging* **11**(2), 206–223 (2002).
11. S. Lyu and H. Farid, "Steganalysis using higher-order image statistics," *IEEE Trans. Inf. Forens. Secur.* **1**(1), 111–119 (2006).
12. I. Avcıbas, M. Kharrazi, N. Memon, and B. Sankur, "Image steganalysis with binary similarity measures," *J. Appl. Signal Process.* **17**, 2749–2757 (2005).
13. T. Ojala, M. Pietikainen, and D. Harwood, "A comparative study of texture measures with classification based on feature distributions," *Pattern Recogn.* **29**, 51–59 (1996).
14. P. Pudil, F. J. Ferri, J. Novovicov, and J. Kittler, "Floating search methods for feature selection with nonmonotonic criterion functions," in *Proc. 12th IEEE Int. Conf. on Pattern Recognition*, Vol. 2, pp. 279–283 (1994).
15. A. C. Rencher, *Methods of Multivariate Analysis*, Wiley, New York (1995).
16. P. H. A. Sneath and R. R. Sokal, *Numerical Taxonomy: The Principles and Practice of Numerical Classification*, W. H. Freeman, San Francisco, CA, (1973).
17. O. Celiktutan, B. Sankur., I. Avcıbaş, and N. Memon, "Source cell phone identification," in *Proc. 13th Int. Conf. on Advanced Computing & Communication—ADCOM 2005*, ACS, pp. 1–3 (2005).
18. www.adobe.com.
19. B. Gokberk, H. Dutagaci, L. Akarun, and B. Sankur, "Representation plurality and decision level fusion for 3D face recognition" (submitted for publication).
20. D. Andreutos, K. N. Plataniotis, and A. N. Venetsanopoulos, "Distance measures for color image retrieval," in *Proc. IEEE Int. Conf. on Image Processing*, Vol. 2, pp. 770–774, IEEE Signal Processing Society (1998).
21. A. M. Eskicioğlu, "Application of multidimensional quality measures to reconstructed medical images," *Opt. Eng.* **35**(3), 778–785 (1996).
22. N. B. Nill, "A visual model weighted cosine transform for image compression and quality assessment," *IEEE Trans. Commun.*, **33**(6), 551–557 (1985).



Nasir Memon is a professor in the Computer Science Department at Polytechnic University, New York. He is the director of the Information Systems and Internet Security (ISIS) lab at Polytechnic University. His research interests include data compression, computer and network security, digital forensics, and multimedia data security.

Biographies and photographs of the other authors not available.

IMPROVEMENTS ON SENSOR NOISE BASED SOURCE CAMERA IDENTIFICATION (EXTENDED ABSTRACT)

Y. Sutcu, S. Bayram

Polytechnic University
Electrical & Computer Engineering Dept.
Brooklyn, NY, 11201, USA

H. T. Sencar, N. Memon

Polytechnic University
Computer & Information Science Dept.
Brooklyn, NY, 11201, USA

ABSTRACT

In [1], a novel method for identifying the source camera of a digital image. The method is based on verifying the presence of digital cameras sensor's pattern noise in a digital image through a correlative procedure. In this paper, we investigate the performance of this method in a more realistic setting and provide results concerning its detection performance. To improve the applicability of the method as a forensic tool, we propose an enhancement over it by also verifying that class properties of the image in question are in agreement with those of the camera. For this purpose, we identify and compare characteristics due to demosaicing operation. Our results show that the enhanced method offers a significant improvement in the performance.

Index Terms— Digital Forensics, source camera identification, pattern Noise, demosaicing artifacts

1. INTRODUCTION

Digital imagery is becoming an integral part of our daily lives at a rapid pace. As a result of this shift in technology, conventional film photography is disappearing. When combined with the availability of extremely powerful image processing techniques and computer graphics technologies, this trend poses new issues and challenges concerning the authenticity and integrity of digital images. This problem is further exacerbated when photographic evidence is considered. Digital image forensics techniques aim at closing this gap by uncovering facts about a digital image. Due to wide popularity of digital cameras, design of many digital image forensics techniques requires an understanding of the fundamental operation of the digital camera.

In this regard, the core element of a digital camera is a charged coupled device (CCD) which measures the intensity of light incident on it. A CCD sensor is essentially a two-dimensional array of light sensitive elements called pixels. Similar to other electronic devices, a CCD sensor is also subject to measurement noise [1]. More specifically, the noise in a digital image can be assumed to have two main components: (1) the shot noise which is assumed to be a random component, and (2) the pattern noise which is assumed to be a deterministic component. Furthermore, the pattern noise consists of two main components which are the fixed pattern noise (FPN) and the photo-response non-uniformity noise (PRNU).

One component of the fixed pattern noise (FPN) is due to dark currents which refers to pixel-to-pixel differences when the sensor array is not exposed to light. Dark current noise can be easily compensated within a camera by taking dark frame and subtracting it from a sensor output. On the other hand, the other part of the pattern noise, photo-response non-uniformity noise (PRNU), is primarily caused by sensitivity of pixels to light and it is primarily caused by the imperfections in the sensor manufacturing process. However, unlike dark currents PRNU cannot be easily corrected. Therefore, uniqueness of the photo-response non-uniformity noise makes it a compelling means for characterizing digital cameras.

In [1], authors proposed a method to extract the pixel non-uniformity noise associated with a CCD sensor. The key idea of the method is to denoise the image by a wavelet-based denoising algorithm so that the resulting residue contains the needed noise components. However, since the underlying image model used in denoising is an idealistic one the residue signal also contains contributions from the image signal. Hence, to eliminate random component of the noise denoising is applied to a set of images (captured by the same camera) and the corresponding noise residues are averaged to obtain the *reference pattern* of a given digital camera. Later, to determine whether a given image is captured by a digital camera, the noise pattern extracted from the individual image is correlated with the reference pattern of the digital camera. A decision is made based by comparing the measured correlation statistic to a pre-determined decision threshold. Figure 1 illustrates the steps involved in matching an image to a digital camera.

Since each CCD element (pixel) is essentially monochromatic, capturing color images requires separate CCD arrays for each color component. However, due to cost considerations most digital cameras use only a single CCD sensor along

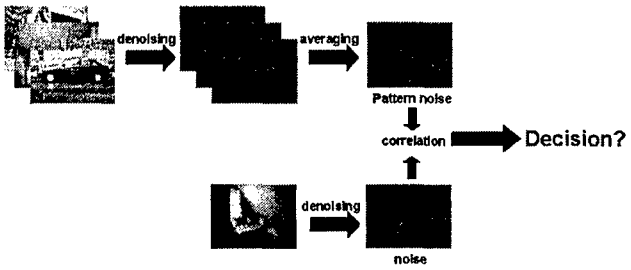


Fig. 1. Illustration of the Sensor Pattern Noise Based Source Identification Method Proposed in [1]

with an array of spectral filters in front of the sensor, namely, color filter array (CFA). The CFA essentially arranges pixels in a pattern so that each pixel captures one of the red, green or blue colors, and the missing color values for each pixel is later computed through a process called demosaicing, a form of interpolation that uses color information from neighboring pixels to obtain the color value of a pixel. At the same time, however, demosaicing operation (interpolation) introduces pixel-wise correlations whose specific form depends on the specifics of the interpolation. In [2, 3], we utilize such artifacts to identify the source camera-model of a digital image. Since, extraction of a reference pattern noise of a digital camera requires the availability of a number of images (taken by the same camera), this approach can be incorporated to sensor noise based source camera identification by also characterizing the underlying demosaicing operation and verifying that the test image also exhibits similar characteristics. Such a combined test will improve the accuracy in the matching process since in addition to individual camera properties camera's class properties are also involved in the decision.

The rest of the paper is organized as follows. In Section 2, we discuss the limitations of the methodology described in [1] to be applied in a forensic investigation and present performance results obtained under a more realistic scenario. In Section 3, the potential of demosaicing artifacts [2, 3] in identifying the class properties of camera (e.g., camera-model) from a given set of images is discussed and experimental results are presented. Incorporation of the demosaicing artifacts to sensor noise based source camera identification method and the corresponding performance results are given in Section 4. Our conclusions and future efforts are given in Section 5.

2. PATTERN NOISE AND SOURCE CAMERA IDENTIFICATION

In [1], the authors carry out the performance analysis (e.g., detection and false-positive rates) by considering pairwise comparisons since the corresponding (pairwise) distributions can be well modeled by the generalized Gaussian. I.e., when the test data is constructed by pooling all images taken by other digital cameras together the distribution no longer follows the model. However, this approach is not preferable when real-life forensics analysis is considered as it does not yield to true false-positive (false-acceptance) rates, which is one of the most important parameters of a forensics method. Therefore, in our setup, we rather compared the correlation values obtained from images taken by a given camera with the correlation values calculated from a mixed set of images. (It should also be noted that in our experimental setup there is no overlap between the training and test image sets.) The performance results obtained under this setting are given in Figure 2. It can be observed that the overall performance of the method is found to be worse than the reported results in [1] due to differences in the experiments. However, in the next section, we describe a mechanism to improve the false-positive versus true-positive detection performance of the sensor noise based source camera identification technique.

In our experiments, we considered three different digital cameras. These cameras include a Sony DSC 90, a Sony DSC 72p and a Canon Powershot S1 IS. Number of images taken by these three cameras are 1214, 894 and 944, respectively. The images in all sets are in JPEG format and are of sizes 1728×2304, 960×1280, and 1536×2048, respectively. In each set, 300 images are selected randomly (as the training set) and used in extraction of the reference pattern of each camera. The rest of the images are used for testing and verification purposes.

In order to extract the noise patterns, we implemented the very same denoising filter employed in [1]. The performance results for the three cameras are obtained in the following manner:

- The reference patterns are obtained by averaging the 300 noise residual signals from the training sets of each camera.
- Denoising algorithm is applied to each of the test images and the extracted noise is correlated with the reference pattern. (It should be noted that 300 training images are not used in this step.)
- Denoising algorithm is applied to a set of approximately $50K$ images. Noise residuals extracted from these images are also correlated with the sensor pattern noise of each camera. (In cases when the size of the noise image is different from the size of the sensor noise pattern, the larger one is cropped appropriately to match the smaller one.)

As a result of the last two steps, we obtain two set of correlation values for each camera. The distribution of two detection statistics are then used to obtain the receiver operating characteristic (ROC) curves in terms of false rejection rate (FRR) and false acceptance rate (FRR) values, as given in Figure 2.

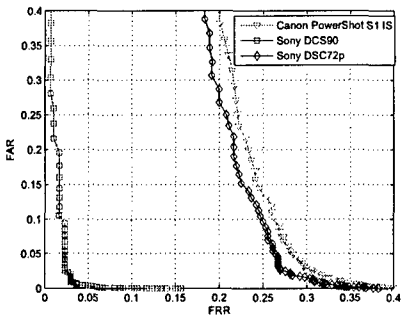


Fig. 2. ROC curves for three different cameras

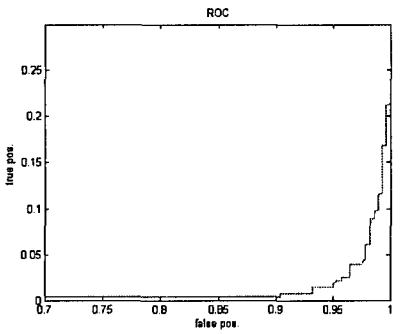


Fig. 4. Performance of the proposed method for Sony DSC S90

It can be observed from these performance results that, although one of the three digital cameras can be identified more successfully than the others, the measured FAR and FRR values are not satisfactory enough for the method to be regarded as as a reliable forensic tool.

It should be emphasized that if the test images also include the training set (i.e., the 300 images), the correlation values would yield a better differentiation between the test images and the randomly generated set (i.e., 50K images). Figure 3 shows the improvement in the correlation values when the reference pattern noise extracted from a camera is correlated with the noise extracted from images in training set as compared to images in the test set. While the mean value of the correlations obtained from the set of test images is 0.08, the mean calculated over the set of training images is 0.12.

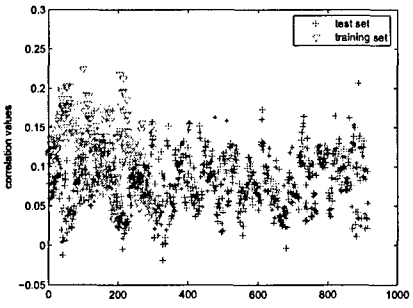


Fig. 3. Comparison of the correlation values obtained from test and training datasets for Sony DCS90

3. DETECTING INTERPOLATION ARTIFACTS

In [2, 3], we proposed a source camera-model identification method based on the observation that interpolation operation will introduce artifacts in the form of periodic correlations among image pixels [4]. The specific form of the the artifacts depend on the specifics of the demosaicing algorithm and they can be used to classify the images as originating from a certain class of digital cameras We have reported the results of this multi-class decision process in [2, 3]. In this paper, we reduce the problem to a single-class classification problem to decide whether an image is taken by a particular digital camera model or not. In other words, rather than trying to categorize the structure of artifacts, we determine if the artifacts exhibit a specific structure. Since in the feature space, features associated with all digital cameras will be more dispersed as compared to those of a single camera, the difficulty of classification will be lower thereby yielding an improvement in the performance.

In our initial experiments we deployed the images taken by the Sony DSC S90 camera and extracted the features from the training set (e.g., 300 images). In a similar manner, we extracted features from a mixed set of 2K images captured by various models of digital cameras. In the first step of our experiments, we used a one-class SVM classifier to decide whether or not an image is taken by a Sony DSC S90 camera. For this, we used 100 images, out of 300 images, to train the classifier. Then, we tested the constructed classifier on remaining 200 images and the images from the mixed set. The resulting accuracy is computed as 87.8%. To overcome the limitations of one-class classifier design (in obtaining the decision hyper-plane), in the second set of our experiments, we considered two-class classification. Hence, in addition to 100 images from the camera training set we included 100 images from the mixed set in designing the classifier, where the remaining $200 + 1900$ images are used for the testing the resulting classifier. In this case, the performance increased to 96.6% and the corresponding ROC is shown in Figure 4.

These results show that demosaicing artifacts give better results when used with single-class approach. Furthermore, in the next section we show how to use the same approach to reduce false-acceptance rate of the sensor noise based source camera identification method.

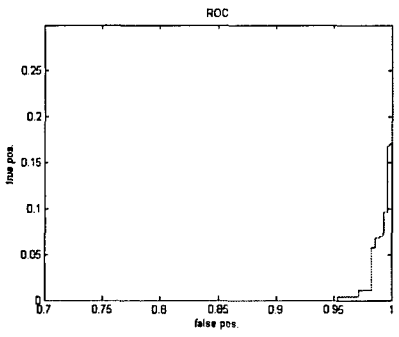


Fig. 5. Performance of the combined decision process for Sony DSC s90

4. COMBINING PATTERN NOISE PROPERTIES WITH DEMOSAICING CHARACTERISTICS

To improve the accuracy of sensor noise based source camera identification technique described in Section 2 we enhance it by also verifying the consistency of demosaicing artifacts, as described in Section 3. This can be realized by one of the following two methods.

4.1. Combined Decision Process

In this case, the decision is the result of a combined decision mechanism wherein the features concerning demosaicing artifacts are combined with the measured correlation value into a single feature vector and a classifier is designed accordingly. In the experiments, we used the same image set as in Section 3. The accuracy obtained for Sony DSC S90 camera was 98.21%. The corresponding ROC curve is given in Figure 5. The improvement in the classification accuracy indicates that the information carried by demosaicing artifacts and the correlation value are complementary in nature.

4.2. Cascaded Decision Process

In this case, the two methods are used in sequence before making a decision. In the first round, match of sensor noise is checked and then, if the result is positive, secondary features are analyzed. Therefore in the experiments, at first positive decisions due to pattern noise matching (true positive and false negatives) are determined. Then, these images are feed to the classifier to verify the consistency of demosaicing artifacts. Hence, the final decision is made by the classifier. It should be noted that in this setting the use of a classifier may eliminate some of the false positives while at the same time reducing some of the true detections as well. Therefore, the question to be answered is if the decrease in false-positive rate can compensate for the decrease in the true detection rate. The results corresponding cascaded decision process is given

	Pattern Noise Match		Classification Using Periodicity Features	
Decision Threshold	True Positive (%)	False Positive (%)	True Positive (%)	False Positive (%)
0	71.33	48.84	69	0.58
0.002	60.67	48.26	58.67	0.58
0.005	43.33	18.02	42	0.29

Fig. 6. Performance of the combined decision process for Sony DSC s90

in Figure 6. This shows that cascaded decision process offers viable alternative as the false-positive rate decreases considerably whereas the true detection rate remains almost the same.

5. ONGOING WORK

In this paper, we propose an improvement over source camera identification based on sensor's pattern noise. Our method is motivated by the observation that when the reference pattern of a digital camera is correlated by the noise extracted from many images, as initially proposed in [1], the resulting false positives are more than estimated by the numerical computations. To address this problem, we propose a scheme that enables application of this method in a more realistic forensics scenario. This is realized by incorporating the digital camera's demosaicing characteristics into the decision process thereby increasing the reliability of the decision. Preliminary results show that we are able to reduce false-acceptance rate of the sensor pattern noise method. In the final version of the paper, further results obtained on other digital cameras and on a larger mixed dataset will be provided.

6. REFERENCES

[1] Jan Lukas, Jessica Fridrich, and Miroslav Goljan, "Digital camera identification from sensor noise," *IEEE Transactions on Information Security and Forensics*, vol. 1(2), pp. 205–214, June 2006.

[2] S. Bayram, H. T. Sencar, N. Memon, and I. Avcibas, "Source camera identification based on cfa interpolation," *Proc. IEEE International Conference on Image Processing ICIP 2005*, 2005.

[3] S. Bayram, H. T. Sencar, and N. Memon, "Improvements on source camera-model identification based on cfa interpolation," *Proc. of WG 11.9 International Conference on Digital Forensics*, 2006.

[4] A.C. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of re-sampling," *IEEE Transactions on Signal Processing*, vol. 53(2), pp. 758–767, 2005.